Photogrammetric Computer Vision Methods for the Automatic 3D Reconstruction of Virtual Sports Environments

by Konstantinos Naskou Athanasios

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy, August 2018



University of
Nottingham
UK | CHINA | MALAYSIA

Contents

List of Figures v

List of Tables x

I Prologue 1

Abstract 2

Acknowledgements 4

1 Introduction 5

1.1 Motivation for the Study ... 7

1.2 Aim and Scope ... 12

1.3 Objectives ... 13

1.4 Contributions ... 15

1.5 Thesis Structure ... 16

II Status Quo 18

2 Virtual Environments in Sliding Sports Training 20

2.1 Combined Bobsleigh, Skeleton, and Luge Tracks ... 22

2.2 Winter Sports Simulation ... 28

2.2.1 Track-Sled-Driver Interaction ... 30

2.2.2 Mathematical Representation of the Track's Surface ... 31

2.3 Whistler Track Construction Study ... 33

Chapter Summary ... 35

3 Photogrammetric Computer Vision 36

3.1 The Photogrammetric Process ... 38

3.2 Project Planning ... 40

3.2.1 Site Constraints ... 41

3.2.2 Measurement Mode ... 43

3.2.3 Surface Matters ... 46

3.3 Estimation Theory and Practice ... 48

3.3.1 Self-Calibrating Bundle Adjustment ... 50

3.3.2 Feature Detection, Description, and Matching ... 53

3.3.3 Dense Image Matching ... 56
3.4 Photogrammetric Network Design ... 58
3.5 Segmentation and Recognition ... 61
3.5.1 Image Segmentation ... 62
3.5.2 Three-Dimensional Shape Detection ... 66
3.6 State-of-the-Art Software and Hardware Implementations ... 67
3.6.1 Image-Based 3D Scanning Software ... 67
3.6.2 Hand-Held Lasers Scanners ... 72
Chapter Summary ... 75

4 Geometry Processing **76**
4.1 Geometric Shape Representation and Optimisation ... 78
4.2 Curve Fitting ... 82
4.2.1 Curvature Estimation ... 85
4.2.2 Alignment Segmentation and Extraction ... 87
4.3 Surface Fitting ... 92
Chapter Summary ... 95

5 Preliminary Investigations **96**
5.1 Whistler Track Reconstruction from a Track Walk Video ... 96
5.2 Winterberg Track Reconstruction from Track Walk Images ... 100
Chapter Summary ... 102

6 Synopsis **103**

III Episteme **109**

7 Image-Based 3D Scanning **111**
7.1 Network Design Simulations ... 111
7.1.1 Materials and Methods ... 112
7.1.2 Simulations on a Cylindrical Surface ... 119
7.1.3 Simulations on the 3D Model of a Track Segment ... 124
7.1.4 Simulations on a Track Segment ... 128
7.2 Track Image Segmentation ... 131
7.3 Track Start Line Detection ... 137
Chapter Summary ... 142

8 Geometric Scene Reconstruction **145**
8.1 Track Alignment and Datum Definition ... 145
8.1.1 Initialising the Track's 3D Alignment ... 147
8.1.2 Optimising the Datum Definition ... 148

8.2 Horizontal Alignment Extraction ... 152
8.2.1 Curvature-Based Alignment Segmentation ... 153
8.2.2 Control Polygon Extraction ... 157
8.2.3 Clothoid Spline Fitting ... 159
8.2.4 Experimental Results ... 161
8.3 Track Surface Approximation ... 167
8.3.1 Fitting the 3D Track Alignment ... 169
8.3.2 Cross-Sectional Curve Segmentation ... 173
8.3.3 Sweeping and Morphing ... 176
Chapter Summary ... 183

IV Synthesis **184**

9 The Proposed Workflow **186**
9.1 Resources ... 186
9.2 Data Collection ... 187
9.3 Data Processing ... 188
Chapter Summary ... 189

10 Lillehammer Track Scanning **191**
10.1 Image-Based Scanning and Sparse Reconstruction ... 191
10.1.1 Reference measurements ... 192
10.1.2 Processing protocol ... 194
10.1.3 Image Orientation ... 194
10.2 Online Feedback ... 196
10.3 Comparisons with Alternative Data Sources ... 201
10.3.1 Design Plans ... 201
10.3.2 Orthoimagery ... 201
10.3.3 Hand-Held Laser Scanning ... 202
10.3.4 Quality Assessment ... 205
10.4 Track Modelling ... 207
Chapter Summary ... 212

11 Igls Track Scanning **216**
11.1 Concrete Track Scanning ... 216
11.2 Ice Track Scanning ... 219
11.2.1 Image Orientation ... 220
11.2.2 Image Segmentation ... 222
11.2.3 Dense Reconstruction ... 225
11.3 Automatic Track Surface Extraction and 3D Modelling ... 232
Chapter Summary ... 239

V Epilogue **240**

12 Discussion **242**

13 Conclusions **250**

Bibliography **253**

VI Appendices **287**

A Preliminary Trials **288**

B Network Design Simulation Scatter Plots **291**

C Image Sequence Quality Reports **294**

D Camera Calibration **305**

E GNSS RTK Survey Reports **309**

F Lillehammer Case Study **311**

G Igls Case Study **318**

List of Figures

1.1 A north-east oblique view of the Lillehammer track. 10
1.2 Schematic illustration of the main objectives within the automatic track reconstruction workflow. 17
II.1 Virtual environment creation pipeline. 19

2.1 Forces acting on a sled during a run down the track. 21
2.2 International competition hosting tracks. 23
2.3 The typical starting area and timing system of a sliding track. 26
2.4 Whistler track geometry. 27
2.5 Track surface preparation. 28
2.6 Track surface parameterisation. 32

3.1 A metamodel for automated photogrammetric computer vision systems 39
3.2 Thick lens model. 41
3.3 Relationship between object size, accuracy requirements, and measurement method. 42
3.4 Materials used in sliding track construction. 48
3.5 Photogrammetric coordinate systems. 51
3.6 Feature point detection and matching. 55
3.7 Epipolar geometry. 57
3.8 Network design simulations on objects of standard geometry. 60
3.9 SLIC superpixels and graph-based image segmentation. 65
3.10 The typical image-based 3D modelling workflow in Agisoft PhotoScan. 69
3.11 The Zeb1 hand-held laser scanner. 72
3.12 Top view of the tunnels at the Camborne School of Mines test site. 74

4.1 Workflows for the 3D reconstruction and modelling of a sliding track's surface. 77
4.2 Examples of non-manifold triangle mesh vertices and edges. 78
4.3 A non-rigid curve registration example using the Coherent Point Drift algorithm. 80
4.4 Clothoid curves and their curvature profile. 83
4.5 Schematic surface reconstruction. 94
4.6 Detection and reconstruction of free-form sweeps. 94
4.7 Morfit: interactive surface reconstruction. 94
4.8 NURBS fitting for as-built modelling. 95

5.1 Camcorder trajectory and video frames extracted at 5 fps along the starting straight of the Whistler track. 97
5.2 Blocked views of the Whistler track. 97
5.3 Image quality estimation for three blurry video frames of the Whistler track. 98
5.4 Dense reconstruction of the Whistler track. 99
5.5 Along- and across-track distances between stereo pairs with respect to their position along the first 550 m of the Winterberg track. 100
5.6 Sparse point clouds of three disjoint camera networks of the Winterberg track. 101
5.7 Perspective view of the dense point cloud and stereo pairs along the starting straight of the Winterberg track. 101

7.1 Examples of the standard imaging configuration for the stereo-normal case. 112
7.2 Schematic representation of the imaging configurations for the 25 generic blocks which are used in the network design simulations. 115
7.3 Schematic representations of the geometric properties of a generic block. 116
7.4 The yaw rotation of the left and right cameras of the stereo pairs which were collected along a segment of the Winterberg track. 117
7.5 Two diverging stereo pairs collected at the Winterberg track. 118
7.6 The texture map that was applied on the cylinder object which was used in the first set of the network design simulations. 120
7.7 The test object that was used in the first set of camera network design simulations. . 121
7.8 Rendered views of the cylinder that was used in the network design simulations. 123
7.9 Boxplots of the total RMS of the feature points' radial distance errors, which were estimated from the self-calibrating bundle adjustment simulations. 125
7.10 Boxplots of the total RMSE of the coded target coordinates, which were estimated from the self-calibrating bundle adjustment simulations. 126
7.11 The test scene that was used in the second network design simulation experiment. .. 127
7.12 Network design simulations on a model of bend 3 of the Igls track. 129
7.13 The 3D model of bend 3 of the Igls track. 130
7.14 The imaging configurations that were used for the dense point cloud generation along bend 3 of the Igls track. 132
7.15 Colour-coded dense point clouds of bend 3 of the Igls track, generated with imaging networks comprising generic block 5 or 26. 133
7.16 Colour-coded dense point clouds of bend 3 of the Igls track, generated with imaging networks comprising generic block 11. 134
7.17 Track segmentation on an image of a straight segment. 137
7.18 Track segmentation on an image of a bend. 138
7.19 Track segmentation on two images of the Sochi track, variously affected by misplaced background pixel seeds. 138

7.20 Track segmentation on two images of the Sochi track, variously affected by correctly placed background pixel seeds. 139
7.21 Track segmentation on two images of a part of the Sochi track where not both sidewalls are covered with ice. 139
7.22 The start line at the Sochi, Igls, and Lillehammer track. 141
7.23 Automatic start line detection. 143
7.24 Matching and triangulating the image projections of endpoint candidates for the detection of the Lillehammer track start line. 144

8.1 Identifying pixels which belong to the same cross section of the track without targets is challenging even for a human operator. 146
8.2 Track alignment initialisation by fitting a cubic smoothing spline to the piece-wise linear sensor path. 149
8.3 Approximate transformation of the photogrammetric model of the Whistler track compared to a reference track alignment. 151
8.4 Clothoid spline fitting workflow. 153
8.5 Horizontal alignment extraction - 2nd Experiment: Curvature-based alignment segmentation. 154
8.6 The deflection of a circular arc segment. 155
8.7 Fitting individual bend models in curvature and object space. 156
8.8 A control polygon simplification example. 158
8.9 Estimating the central angle α at a control polygon vertex. 159
8.10 Horizontal alignment extraction - 1st Experiment: Clothoid spline fitting to a uniformly spaced sample. 162
8.11 Horizontal alignment extraction - 1st Experiment: Signed orthogonal distances from the reference curve. 162
8.12 Horizontal alignment extraction - 2nd Experiment: Signed orthogonal distances from the reference curve. 162
8.13 Horizontal alignment extraction - 2nd Experiment: Non-uniformly spaced sample. 164
8.14 Horizontal alignment extraction - 2nd Experiment: Cornucopia (2010) result. 164
8.15 Horizontal alignment extraction - 2nd Experiment: The proposed method on a non-uniformly spaced noisy sample. 165
8.16 Horizontal alignment extraction - 2nd Experiment: AutoCAD Civil 3D (2018) result. 166
8.17 A 3D mesh which represents parts of the Igls track and its surrounding environment, as well as a space curve approximating its centreline alignment. 167
8.18 Extraction of cross-sectional curves along an approximate track alignment. 168
8.19 Examples of typical track cross-sectional curves. 170
8.20 Defining the floor level on a cross-sectional curve and the region of interest for template matching. 170

8.21 Template matching on a cross section that was extracted along the entrance of a right bend. ... 171
8.22 A vertical cross-sectional cutting plane along the alignment. ... 172
8.23 Trimming the cross-sectional curves. ... 174
8.24 Shape-based cross-sectional curve segmentation. ... 174
8.25 CPD-based curve segmentation. ... 175
8.26 Track partitioning for the skinning process. ... 176
8.27 Cross-sectional curve detection, segmentation, and selection. ... 177
8.28 Data-driven generation of the track's characteristic cross-sectional curves. ... 179
8.29 The geometric properties of the trimmed cross-sectional curves. ... 180
8.30 Sweeping-and-morphing along a bend segment. ... 182

9.1 A schematic example of the implementation of generic block 26. ... 187
9.2 Track modelling workflow. ... 190

10.1 Views of the Lillehammer track. ... 192
10.2 Ground control points at the Lillehammer track. ... 193
10.3 Processing protocol for image orientation. ... 195
10.4 Lillehammer track, dataset $Lil350B$: Control and check point coordinate residuals. ... 197
10.5 Lilehammer track, dataset $Vid5fps$: Sparse reconstructions from video frames extracted at 5 fps. ... 198
10.6 Lilehammer track, dataset $Vid5fps_{cba}$: Check point residuals. ... 198
10.7 The along-track distances between the successive camera stations and the stereo base for each image pair at the Lillehammer track. ... 199
10.8 Plan view and vertical profile along the Lillehammer track floor. ... 201
10.9 Lillehammer track topography and along track terrain profile ... 203
10.10 Point clouds that were collected at the starting area of the Lillehammer track at the beginning and finish of a laser scanning survey with the Zeb1. ... 206
10.11 Comparisons between the Lillehammer track alignments that were created from various sources (full length). ... 208
10.12 Comparisons between the Lillehammer track alignments that were created from various sources (first 800 m). ... 209
10.13 The internal accuracy of a 0.5 km long Zeb1 loop ($Zeb1s1$) at the Lillehammer track. ... 210
10.14 The internal accuracy of a 1 km long Zeb1 loop ($Zeb1s3$) at the Lillehammer track. ... 211
10.15 Automatic track alignment and surface extraction from dataset $LilZeb1s4$. ... 213
10.16 Automatic track alignment and surface extraction from dataset $Lil350B_{cba}$. ... 214
10.17 The track surface that was generated from dataset $LilZeb1s4$ ... 215

11.1 A north-east oblique view of the Igls track. ... 217
11.2 Views of the Igls concrete track on the day it was scanned. ... 217

11.3 Plan view of the the Igls track with the ground control point locations. 218
11.4 The implementation of generic block 5 and 11 at the Igls track..... 219
11.5 Views of the Igls ice track..... 219
11.6 The implementation of generic block 26 at the Igls track. 220
11.7 Transitions from open to roofed segments along the Igls track..... 220
11.8 Relative orientation trials for an image set of the Igls ice track. 221
11.9 Bundle adjustment results for the Igls ice track scan..... 224
11.10 Views of the Igls track before and after the image segmentation of dataset $IglsIce26$. 225
11.11 Assessment of the automatic track segmentation process in image space. 226
11.12 Igls track start line detection. 227
11.13 A sparse point cloud of the Igls track and two options for the definition of the bounding box that was used during the 3D point cloud and mesh generation. 228
11.14 Reconstruction of the Igls track surface subdivided in bounding boxes. 231
11.15 Igls track surface reconstruction using the visibility-consistent mesh generation method. 231
11.16 The first five bends of the Igls track, reconstructed using two different mesh reconstruction methods. 233
11.17 Cross sections extracted along bend 7 at the Igls track. 235
11.18 The Igls track surface model that was generated from dataset $IglsConcCN_{cba}$. 236
11.19 Views of the reconstructed 3D mesh and the extracted surface of the Igls (concrete) track for dataset $IglsConcCN_{cba}$ (reference). 236
11.20 The Igls track surface model that was generated from dataset $IglsIce26C_{sim}$. 237
11.21 Views of the reconstructed 3D mesh and the extracted surface model of the Igls (ice) track for dataset $IglsIce26C_{sim}$..... 237
11.22 The Igls track surface model that was generated from dataset $IglsConcC5_{arb}$. 238
11.23 Views of the reconstructed 3D mesh and the extracted surface model of the Igls (concrete) track for dataset $IglsConcC5_{arb}$. 238

12.1 The Igls track model. 248
12.2 Views of the Igls track model. 249

List of Tables

2.1 Geometric properties of combined bobsleigh, skeleton, and luge tracks. 24

7.1 The attributes of the imaging configurations that were used in the network design simulations experiments on a cylindrical surface. 120
7.2 Parameter settings for the image orientation simulations on the 3D mesh of bend 3 at the Igls track. 128
7.3 The results from the network design simulations on a track segment. 131

8.1 Coherent point drift algorithm parameter settings for non-rigid registration of cross-sectional curves. 175

10.1 Parameter settings for the image orientation process of the Lillehammer dataset within PhotoScan. 193
10.2 Lillehammer track, dataset *Lil*350*B*: SfM solution assessment. 196
10.3 Qualitative assessment of the effect of image quality, acquisition geometry, and reflectance properties of the scene on the success of the relative orientation process. 200
10.4 Length and duration for each laser-scan survey with the Zeb1 at the Lillehammer track. 204
10.5 The along-track quadratic average of the orthogonal distances from the assessed space curves to a reference centreline alignment of the Lillehammer track. 207
10.6 Arc radii estimation for the first four corners of the Lillehammer track. 212

11.1 Image orientation report for 10 trials in PhotoScan which successfully recovered the relative orientation of the entire image set of the Igls ice track. 223
11.2 Dense point cloud and mesh sizes and processing times for the Igls track datasets. 229
11.3 Igls track alignment extraction report. 234

Part I

Prologue

Abstract

The scan-to-BIM conversion of 3D point clouds of infrastructure works is essential for their as-built documentation and maintenance; automating this tedious and labour-intensive task is a challenging civil engineering problem. Related work on the subject has provided remarkable results with respect to the automatic detection and 3D modelling of planar, spherical, and cylindrical shapes in laser scans of buildings or industrial sites, e.g. walls and pipes. However, long and narrow spaces comprising free-form surfaces with reflective properties unfavourable to optical measurements pose additional challenges, especially when the precise definition of their boundaries is also required. Image-based 3D scanning techniques, in particular, are more liable to yield noisy and incomplete point clouds that include outliers and clutter, which complicate the 3D modelling process.

This thesis is concerned with the dense 3D reconstruction and parametric modelling of objects with a characteristic elongated geometry that can be represented as generalised cylinders, such as race tracks or roadways. The study is focused on a piece of infrastructure which is essential to a sports engineering application. The aim was to explore the potential of designing a robust workflow for the extraction of the 3D model of an approximately 2.5×1500 m long artificial combined bobsleigh, skeleton, and luge sliding track, which can be carried out by non-specialists in geomatics processes. The geometry of the track's concrete structure and ice layer is of interest, so that it can be used in computer simulations for optimal trajectory identification and track construction verification or within a virtual reality system as an aid to both amateur and elite athletes for site familiarisation, training, performance enhancement and, more importantly, accident prevention.

The emerging field of infrastructure computer vision provides several tool for this purpose, based on the application of computer vision methods for the collection and analysis of real world data in order to convert it to useful information. Some of the problems that it aims to solve overlap with the objectives of this thesis: the provision of guidance for proper data collection, combining existing information with human input and sensor output, separating signal from noise in photogrammetric measurements, compensating for missing data due to the difficulties inherent in dealing with certain materials and surface types, resolving the scale ambiguity in generating 3D information from monocular image-based reconstruction methods, as well as, point cloud segmentation and geometric shape detection and fitting.

The first part of the research that is presented in this thesis is concerned with photogrammetric network design simulations in search of an imaging configuration simple enough to be implemented on site with a hand-held camera and compact enough to allow for image acquisition within the time allocated for a typical track walk. In addition, the potential of facilitating the dense reconstruction process by automating an interactive graph-based image segmentation algorithm using 3D information from the sparse reconstruction phase is investigated. In the second part, an automatic clothoid spline fitting method for horizontal alignment extraction is proposed, in order to support a sweeping-and-morphing algorithm that detects the across-track boundaries of the track's surface and fills in missing information along the track's length.

The main hypothesis was that the image-based 3D scanning of the entire length of an ice track can be successfully completed within the limited time frame of a typical track walk by a single person with a hand-held, high-resolution, non-metric, pre-calibrated DSLR camera with a moderately wide prime lens and that the image orientation could be retrieved automatically, without the requirement for scene signalisation or manual image observations, by low-cost software. The process should allow for the parametric 3D model of the track's surface to be extracted from the generated 3D mesh of the site without requiring user interaction. The proposed techniques were validated on two track sites using data from image-based and laser scanning surveys. It was concluded that this scenario is feasible but the integrity of the result and the repeatability of the process are not guaranteed. Subdecimetre accuracy can be achieved for the track's sparse reconstruction if ground control points are measured, with the typical GNSS RTK technique accuracy, at 100–150 m intervals along the track and used in a constrained self-calibrating bundle adjustment. If only 3 ground control points are available, submetre accuracy is achievable; a result at this accuracy level can be expected from the Zeb1 hand-held laser scanner within the same time frame but with a potential to greatly simplify the 3D modelling process, yet complicate the generation of an immersive virtual environment in terms of visual fidelity, unless it is combined with an image-based scanning process in order to retrieve colour information. The proposed sweeping-and-morphing algorithm can provide an initial solution for the parametric 3D model of the track's surface without requiring user interaction but the result needs refining, particularly around the edges along the transitions in and out of the bends. Specifically for a track that was scanned in the iced state, and especially if the 3D model is to be used in a virtual reality system, sliding simulations for the verification of the result are required.

Acknowledgements

I would like to express my gratitude to my supervisor Dr Martin J. Smith for giving me the opportunity to pursue a PhD and for his support throughout my studies. I would also like to thank my MSc advisor Professor Vasilios Tsioukas for reigniting my passion for research in photogrammetry and express my appreciation for all the support and resources I received from the Engineering and Physical Sciences Research Council (EPSRC) and the English Institute of Sport (EIS); especially to Naomi Stenhouse and Andreas Schmid.

I am grateful to my parents for being the mere reason of my awesomeness' existence and every supreme gift bestowed upon me—and for not considering this to be a jokey comment. Special thanks to Fotis and Eftychia for their unconditional friendship and support, when I needed it the most. Elef, Artemis, and Niko; thank you for helping me put the words "life" and "balance" after all this work during the last years and for not letting me curb my enthusiasm about parks and recreation. I would also like to thank Bertrand, for his good spirits and tech-savviness were inspiring before, after, and during our lunch breaks.

Last, and definitely least, I would like to talk to myself in the third person and thank him. Thanks to old-me for taking small but decisive steps away from my comfort zone, leading to pursuing a PhD, and current-me for completing it—hopefully—without losing my soul. Future-me, I know how much you enjoy scrutinising other people's work and finding typos; there's one intentionally left, especially for you, on page...

Chapter 1

Introduction

Artificial intelligence techniques are being used in engineering applications in order to solve a wide range of challenging problems. Although the foundation of artificial intelligence can be found in philosophy and mathematics, it owes its name to John McCarthy who set its aim during the 1955 Dartmouth summer school research project as that of "making a machine behave in ways that would be called intelligent if a human were so behaving" (McCarthy et al., 2006). The original statement of the problem did not involve any other intelligent entities, except for humans. Significant effort has been made since then towards not only attempting to understand human intelligence but also building systems that think and act humanly and rationally (Russell et al., 1995). One of the subfields of artificial intelligence is the scientific discipline of computer vision which deals with visual perception and envisages to make computers see. Research in this domain started in the late nineteen-sixties—and was expected to be concluded during another summer school—but it was soon realised that finding a solution to the vision problem would require more than one summer (Papert, 1966; Szeliski, 2011).

computer vision

Depth perception and object recognition are of particular interest to the engineering discipline of close-range photogrammetry, as well. In contrast to computer vision, which can only be defined and has developed within a digital context (Granshaw et al., 2015), the history of photogrammetry goes back to the invention of photography itself in the beginning of the nineteenth century (Luhmann et al., 2014). Even though they are both founded upon an overlapping theoretical basis (Granshaw et al., 2015), interaction between the two disciplines has only started in the nineties and was primarily one-way. The recent advancements in software and hardware technology have allowed photogrammetry to take the next step in its digital era towards automation, benefiting from major contributions developed within the computer science community such as preliminary photogrammetric orientation processes and feature-based image matching (Granshaw et al., 2015). On

close-range photogrammetry

interaction

the other hand, albeit solutions to problems such as bundle adjustment and camera calibration have been well studied for a few decades now, the vision community spent quite some time reinventing them (Triggs et al., 1999; Fraser, 2013). The debate about convergence of the two disciplines has started almost from the outset with quite a few notable advocates (Torlegård, 1967; Hartley et al., 1993; Triggs et al., 1999; Förstner, 1993, 2002; Granshaw et al., 2015) but is still current; partly on the account of differences in culture (Triggs et al., 1999) and terminology (Granshaw et al., 2015). The recently coined term of "photogrammetric computer vision" that has started being used in textbooks (Förstner et al., 2016) and university curricula appears indicative of the spirit within which future developments will occur, q.v. chapter 3.

convergence

The main difference between the two disciplines—which prevents, to a certain extent, the prospect of integration—stems from the fact that close-range photogrammetry is addressed to commercial users with specific demands on the accuracy and completeness of the end product, which in turn requires the application of robust and reliable measurement techniques (Granshaw et al., 2015), thorough project planning, use of prior information about the problem, and quality control (Triggs et al., 1999). In contrast, three-dimensional (3D) computer vision scientists have always been free to pursue the most challenging problems without constraints that may limit their creativity and narrow down the solutions' applicability domain (Remondino et al., 2014; Granshaw et al., 2015).

integration

Traditionally, measurement tasks on medium to large size objects have been carried out by professionals trained solely for this purpose; geodesists, photogrammetrists, and land surveyors have come a long way in establishing methodologies that can guarantee the desired level of accuracy and precision in the output. These methods usually involve expensive equipment and require a certain level of expertise from the users in order to practically achieve the theoretically achievable quality. Nevertheless, limitations on the available budget or time for a task may call for more flexible solutions. It can be argued that the motivation behind every recent photogrammetric development has been an exotic need of a non-engineering discipline or science, such as archaeology or medicine, as strictly civil engineering applications have already established measurement techniques or have very strict guidelines to follow, to allow for much experimentation. The driving force behind every innovation in computer vision technology is the wide spectrum of needs dictated by a broad audience comprising everyone using a personal computer, mobile phone, or digital camera. The recent advancements in virtual reality systems and the ever-increasing industrial need for 3D shape processing technologies is driven by a diverse user base ranging from computer games enthusiasts to surface metrology scientists with very different requirements in terms of cost, accuracy,

user base

level of detail, and automation (Hale et al., 2014, p. 15). Somewhere in-between, lies a number of scientific disciplines that have benefited from the plethora of available sensors and hardware with enhanced processing power capabilities combined with software that offer (semi-)automated workflows.

sports engineering

Sport is another field where measurement technology, mechanical engineering, and computer science have found new applications. Monitoring through observations, video recordings, or the lately ubiquitous wearable sensors, facilitates the efforts of athletes and sports scientists as the first step towards performance improvement. On the one hand, sports engineering is pushing the envelope with respect to the acquired knowledge in material science towards the design, development, and production of sports equipment. On the other hand, virtual reality has now become a commonplace in competitive sports training. For the most part, the interaction between the athletes and the equipment is of importance; this is usually the case if the field or track is of simple and fixed geometry, e.g. ball sports courts. In some cases though, the track or field is more than just a boundary; the interaction with it, is essentially the game itself. Hence, the quality of its geometry's digital representation within a virtual environment is of paramount importance.

The high expectations towards realistic graphics and human-machine interaction that were created from the integration of computer games and real life—brought with devices like ©Microsoft Kinect (2010) and ©Oculus Rift (2016)—and the concept of augmented reality were soon extended to virtual reality systems used for sports training. Nevertheless, competitive sports training requires more than high resolution graphics and realistic scenes; they potentially require a digitised reality, in terms of visual fidelity and user interaction. At some point, they will demand more than simply an abstraction of the environment that can only be achieved if the graphics and the physics engine of the system simulating the 3D world is based on accurate geometric data, which ideally have been acquired and can be updated in an automated way. This is an ever-changing field of research that has been receiving a lot of attention lately from various disciplines.

1.1 Motivation for the Study

BIM

In order to facilitate decision making, civil engineers need appropriate tools for 3D scene reconstruction, visualisation, and understanding. Especially now that the application of the building information modelling (BIM) paradigm is in the process of being extensively used by the computer-aided facilities management (CAFM) and the architectural, engineering, and construction (AEC) industries, these tools need to be able to digitally represent an infrastructure work throughout its life-cycle from planning, design, and construction to operation and maintenance (Volk

et al., 2014). Essential to the BIM process is multidisciplinary collaboration within a digital environment. However, no single piece of software can currently handle the whole scan-to-BIM workflow, which involves measuring geometry and visual appearance using geospatial data capturing technologies in order to create semantically rich parametric objects with both geometric and non-geometric attributes (Volk et al., 2014; Tang et al., 2010; Brilakis et al., 2010; Bhatla et al., 2012; Xiong et al., 2013b). Thus, a variety of software tools are required alongside users with a high level of expertise to operate them, as most tasks require user interaction (Tang et al., 2010; Thomson et al., 2015; Dimitrov et al., 2016).

It has been argued that the reluctance of small companies to invest in software and the lack of skilled personnel are the most significant drawbacks for the widespread use of BIM (Dainty et al., 2015). Therefore, the development of robust and reliable algorithms able to facilitate the scan-to-BIM pipeline by reducing the requirements for user interaction is essential and can be expected to provide a boost to productivity similar to that of the field-to-finish workflow in land surveying. Some of the challenges in scanning and modelling infrastructure works are separating signal from noise and fusing the collected data with user input and existing information from various sources in order to compensate for non-observable quantities; all within a framework that prioritises practical over computational efficiency (Brilakis et al., 2010). Dealing with these issues has become one of the main research priorities in civil engineering towards automating the tedious, error-prone, and subjective as-built and as-is documentation of both new and existing structures (Volk et al., 2014; Patraucean et al., 2015). The emerging field of infrastructure computer vision is focused on the application of computer vision methods for the collection and analysis of real world infrastructure data in order to convert it to useful information. For example, monocular 3D reconstruction and automated inspection are key components of high demand in commercial applications such as detecting building materials and monitoring defects and damage (Brilakis et al., 2015).

as-built

infrastructure computer vision

At the beginning of this project, a review of the current literature—aiming at steering the research efforts towards current and significant challenges—revealed quite a few research opportunities in this field, which can be summarised as follows:

research challenges

- combining existing information with human input and sensor output
- separating the signal from the noise in measurements and compensating for missing data, due to the difficulties inherent in dealing with certain materials and surface types
- guaranteeing the repeatability of spatial data capturing methods especially when non-deterministic algorithms, such as the random sample consensus (RANSAC) Fischler et al. (1981), are involved

- point cloud shape fitting, segmentation, and classification
- resolving the scale ambiguity in generating 3D information from monocular image-based reconstruction methods
- providing user guidance for proper data collection within automated workflows with built-in self-diagnosis
- taking into account practical utility together with pure scientific advancement (Moons et al., 2008; Volk et al., 2014; Patraucean et al., 2015; Fathi et al., 2015; Brilakis et al., 2015).

Whereas, in the sciences, an abstraction is well-suited for the development of a universal theory or method, in engineering, a specific application does not only provide a strong motivation, but also defines a framework within which a problem becomes tractable and a practical solution attainable. To our surprise, an application in an unexpected and exciting domain, posing all of the aforementioned pending problems, presented itself.

Despite its ambiguous name, BIM applies not only to buildings but also to other elements comprising the built environment (RICS, 2014). The geometry of infrastructure works whose length is several orders of magnitude greater that their width, such as highways, rail tracks, tunnels, underpasses, and canals, can be described by their alignment. Structures belonging to the first two categories are designed so that they can accommodate the dynamic movement of vehicles at speeds over 100 kilometres per hour; so do bobsleigh tracks, q.v. figure 1.1. [1] motivation

The 3D model of the track's surface is the most important element of the virtual environment of a simulator for winter sliding sports and its geometric quality and appearance should reflect the attributes of the real track (Gibson, 1979, p. 3). According to Mossner et al. (2011, p. 895), the sled's motion estimation is more sensitive to moderate deviations from the actual track geometry than to parameters of the physical model which is used to describe it. A robust reconstruction process is also essential for the track's construction verification (Sabbioni et al., 2016, p. 249). However, a number of simulators for sliding sports training, which have been created by various research institutes, share a common characteristic: the track was modelled based on its design plans and did not reflect the as-built geometry, q.v. section 2.2.

The question "what does one approximate-the theoretical (design) surface or the actual track surface?" was already posed in Hubbard et al. (1989, p. 228), but until recently (Rempfler et al., 2016) the answer has been consistently the same; the as-designed concrete instead of the as-is

[1] henceforth, the artificially-cooled combined bobsleigh, luge, skeleton tracks will be referred to as tracks and the term sliding sports will refer to those three, unless otherwise specified.

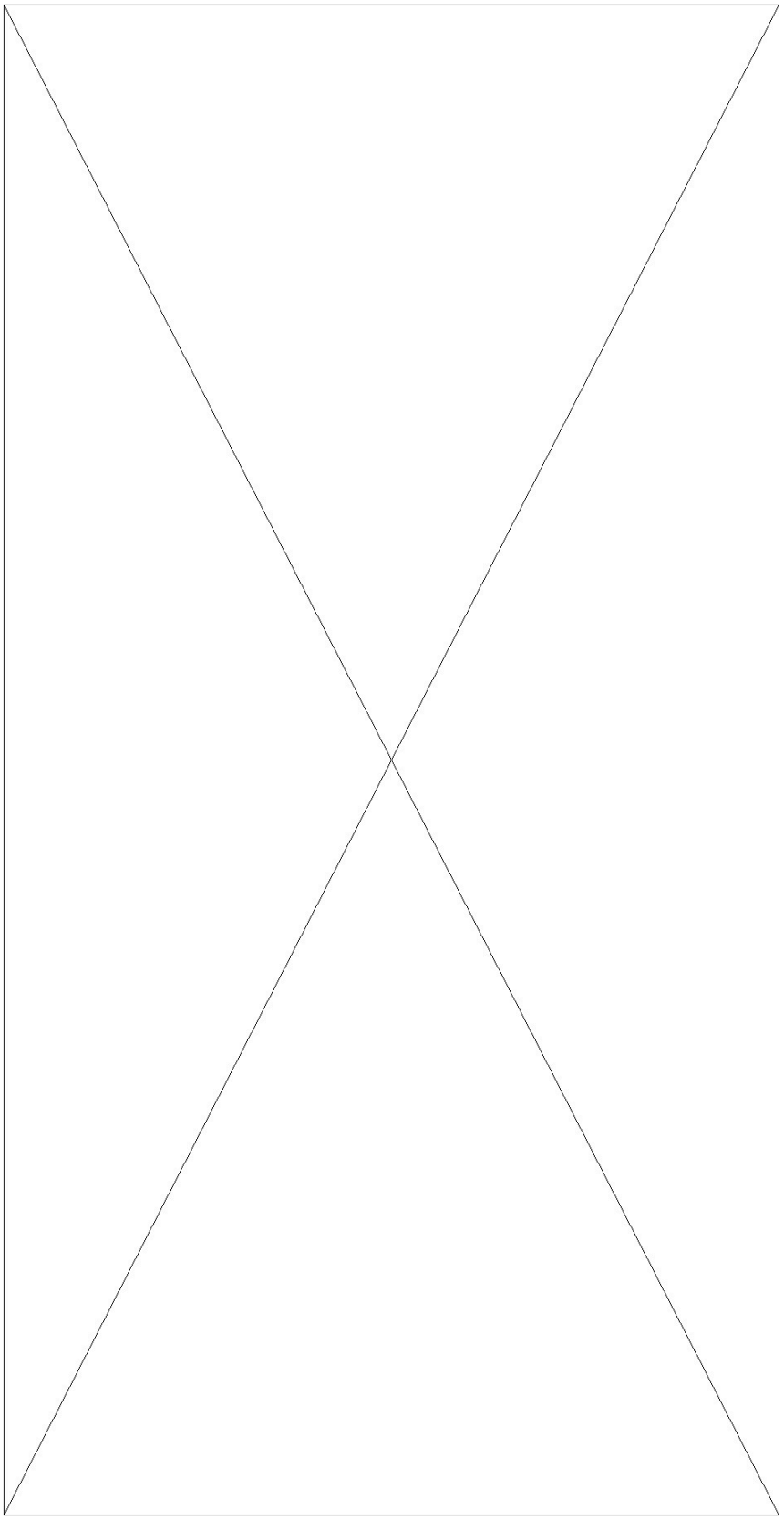


Figure 1.1: A north-east oblique view of the Lillehammer track; reproduced from ©BLOM AS (2017) via Opplysningen (2017). Copyrighted material removed.

ice surface has been considered and potential alterations during construction (IBSF, 2016) were ignored. A notion of realism was added by manual monocular modelling of the surrounding environment based on images of the actual track, which is a tedious process with often questionable results. Nevertheless, even if the surrounding environment and the ice texture can be adequately represented, the forces acting on the sled and the interaction between the ice surface and the runners can only be approximated. This generalisation can act as a desirable smoothing effect that corresponds to the system's resolution, as the task of updating the visual cues from the athlete's perspective and the position and attitude of the sled is computationally intensive. However, the more powerful hardware becomes, the greater the demand for more detailed virtual environments. What is essentially required—and was explicitly requested—by the sport is the necessary tools for a person, e.g. an athlete or a manager, with no experience in surveying, CAD, or 3D modelling to acquire the required information, ideally using an image-based workflow, and digitally reconstruct the track within a few hours.

the question

Reconstructing the as-built 3D model of a track is an undeniably difficult task (Roche et al., 2008, p. 102;Braghin et al., 2010, p. 506;SAIT, 2012, p. 76;Sabbioni et al., 2016, p. 249), as most of the existing approaches are concerned with semi-automatic workflows for scenes comprising planar or cylindrical surfaces (Tang et al., 2010, p. 840; Xiong et al., 2013b; Jung et al., 2014; Son et al., 2015, p. 180; Dimitrov et al., 2015, p. 32; Bosché et al., 2015) and full automation is considered "highly unlikely to be achieved economically in practice" (Brilakis et al., 2010, p. 461). In addition, as Fraser (2013) puts it, "photogrammetry is 'not at its best when measuring long, thin objects'". Such a niche application, however unique a combination of challenges it presents, gave the opportunity to address open problems and revisit some of the difficult tasks in photogrammetry—which are considered to be solved—within unexplored settings. A more than one kilometre long narrow concrete structure, partially roofed, whose surface is covered by layers of ice needs to be surveyed—including its immediate environment—from within its volume within a limited time frame; both passive and active optical methods will struggle to yield results of acceptable quality in terms of accuracy and completeness due to the size and shape of the object and the reflectance properties of its surface. In the absence of artificial targets, which can indirectly define the datum and constrain the geometry of the solution, the reliable definition of the scale and rotation of the reconstructed model with respect to the gravity vector is also a problem. The output which is required for the simulator's physics engine calculations, comprises a 3D parametric model of the track; for the simulator's graphics engine, a polygonal mesh of the surrounding environment needs also be reconstructed.

the problem

The correspondence of the problem-specific challenges to those in the aforementioned list, except for the last one, is straightforward and will be established in sections 1.2 and 1.3, which describe the aim and objectives of this study, respectively. The practical importance of the study to the winter sports community was apparent from the outset. In sliding sports, the gold medal can be decided in a matter of tenths of a second (Hubbard et al., 1989). With the intense competition in elite and Olympic sports and the technology advancements in simulation systems, virtual training is a commonplace between the top ranking teams. Training for track familiarisation is essential to both amateur and elite athletes for performance enhancement but more importantly for accident prevention (Levy et al., 2007). However, there is a limited number of tracks around the world, q.v. section 2.1, mainly due to the cost that is involved in building and maintaining them, especially in unfavourable climates. In addition, the financial requirements for the transportation of the bulky equipment means that the time allocated to each athlete for training on a real track is limited to a couple of hours a year, if any at all (Huffman et al., 1996; Kelly et al., 2000; Rempfler et al., 2016), which prevents teams from tropical or poor countries to even get involved with the sport, let alone train and excel. In fact, the 2018 Nigerian women's bobsleigh team was the first one from an African country to even attempt to qualify, let alone compete, in the Winter Olympics (Pyeongchang, 2018). Putting together a complete simulation system might still be out of reach for most teams but providing the means to create the virtual environment of any given track after a short visit to the site, is a good start and sufficient for virtual tours for track familiarisation.

significance

1.2 Aim and Scope

In broad terms, the aim was to investigate the accuracy potential of a photogrammetric surveying routine embedded in a task as simple as walking along a path, under some quite restrictive conditions related to the nature of sliding sports. The question was can a dataset that was collected in a way that almost guarantees the existence of geometric anomalies in the outcome—due to the reflective properties of ice and the weakly convergent imaging configurations—be processed with minimal interaction and supervision and yield a reliable 3D model of a track? More specifically, the aim of this project was to explore the potential to design a robust, self-diagnosing, automatic workflow for the image-based extraction of the parametric 3D model of a sliding track, which can be carried out by non-experts within a limited time frame, using low-cost, off-the-shelf, hand-held equipment. This entailed assessing and/or improving existing methods on several aspects of the entire process from project planning, data collection, and image processing to point cloud generation, scene reconstruction, and 3D modelling including the quality assessment of the result.

aim

scope

A traditional topographic survey, q.v. section 2.3, is not an option as a number of sport-related constraints are imposed on the surveying approach—primarily due to limited accessibility to the site. The limited space on the track—which is shared with other athletes, managers, and maintenance workers—prevents the use of tripods and other static equipment which, in any case, would require many set-ups due to short lines of sight and the obstruction caused. Hence, a hand-held sensor was considered as more appropriate and desirable from the users' point of view as an inconspicuous means of maintaining a competitive advantage. The image-based approach for the 3D reconstruction was deemed a perfect candidate from the outset, primarily due to the fact that cameras are already extensively used by athletes and trainers in sliding sports during training and official races. A common practice before each training session or race is conducting a track walk in order to assess the condition of the track. Images and video streams of the athletes sliding down the track, and of the track itself during the track walks, are recorded in order to correct errors and plan the next runs.

Ideally, the track's ice surface should be scanned a few hours before a competition and the generation of its 3D model should follow shortly thereafter. Data collection must be concluded within a couple of hours so that it corresponds as much as possible to a coherent snapshot of the constantly changing ice surface and because this is the average time allocated for track walks before a race or during training sessions. The level of detail of the computer generated 3D models is bounded by the scale at which the interaction between the sled and the track's surface takes place, q.v. section 3.2.3. Nonetheless, the limits of the natural laws of physics must be respected; translucent material and specular surfaces are notoriously hard to scan using optical techniques; special methods for this purpose are in their infancy and definitely not suitable for objects of the size of a track (Ihrke et al., 2010). Moreover, no signalised ground control points can be used, primarily due to time limitations and because interference with the track, and the ice surface in particular, is not allowed. In addition, the survey must be conducted in an environment where global navigation satellite systems (GNSS) may not work, as most part of the track is covered with roofs and blinds for protection from the weather.

1.3 Objectives

The objectives of this thesis are the following:

Objective №1 Provide an understanding of the limitations of image-based and laser scanning techniques for track reconstruction using hand-held sensors.

Objective №2 Quantify the precision and accuracy potential of imaging configurations that can be implemented with a hand-held, forward-facing camera for track scanning.

Objective №3 Develop a surface approximation algorithm that can automatically extract the track from a 3D mesh with noise, clutter, and missing information.

Objective №4 Develop an automated workflow for fitting a clothoid spline—which adheres to the design intent of a typical horizontal centreline alignment of a road/track—to a noisy, incomplete, non-uniformly spaced, ordered 2D point set.

Objective №5 Assess the potential of segmenting the images of the track, so that the part of its surface that accommodates the sleds' motion within a virtual reality system can be reconstructed separately from the surrounding environment, in order to better control the point density, mesh resolution, texture quality, and processing effort.

Objective №6 Examine the extent to which certain metrics— which can be calculated online with a small computational overhead for each image—correlate with the integrity of the track's reconstruction, in order to assess whether they can be used to provide feedback to the user during data acquisition.

The intent of the first objective is to gain a theoretical understanding of the, not very well documented, accuracy potential of high-resolution images from a forward-facing camera for dense reconstruction. To this end, a set of simple imaging configurations will be examined using computer simulations on real and computer generated scenes. The theoretical and practical effects of each imaging network design on the quality of the result will be assessed in order to establish a basis for the next steps of the image-based reconstruction process, q.v. figure 1.2a. Some of the questions that will have to be answered are how many images, of what resolution, and in what configuration are required in order to complete the task within the given time frame.

Even if the relative orientation of the camera network can be estimated from features that were detected and matched on the surrounding environment, depth information on the ice surface may be sparse, q.v. figure 1.2c. In addition, the points that are measured on the part of the track that is covered with ice need to be separated from the surrounding environment so that its geometric and appearance attributes can be modelled within a virtual reality system. Existing state-of-the-art approaches for image segmentation, q.v. figure 1.2b, as well as curve and surface fitting, q.v. figures 1.2d and 1.2e, which rely on user interaction or other preconditions, will be adapted and extended to the automation requirements of track scanning by exploiting prior knowledge and by

anticipating anomalies in the data. In doing so, the use of existing software and the development of novel prototypes will be required, not only for the automation and evaluation tasks, but also to create new functionality where it is deemed essential towards the overall aim of the project. As this is way too complicated a process to be thoroughly examined within a partial or purely theoretical framework, the proposed methods will be tested on case studies for the as-built reconstruction of the full length of existing tracks in order to establish a benchmark for future improvements. The performance of a new generation of survey-grade hand-held laser scanners will also be assessed. More details on the methodology that will be used in order to fulfil each objective are provided in chapter 6, after reviewing the related literature in part II.

1.4 Contributions

The overall workflow for the automatic generation of a digital representation of the geometry of a virtual environment for sliding sports training—and the extensive literature review it was based upon—is considered to be the main contribution of this study to the field of sports engineering. It is a well-known fact that, if results of very high accuracy are required, close-range photogrammetry is not appropriate for the reconstruction of long and narrow corridors with low textured surfaces, especially without using a survey control network. Nevertheless, the ubiquitous low-cost image-based scanning (software or hardware) systems that are targeted to non-specialists do not often come with such disclaimers. In addition, any insights with respect to the reconstruction quality that can be achieved for a new scanning project can only be reliably inferred, in order to inform decisions related to its use within a subsequent process, from results on objects with similar geometric and reflectance properties. The case studies that are part of this thesis provide valuable benchmarks that can contain the expectations of track scanning projects with realistic limits; they can also facilitate well-informed decisions when it comes to designing the related virtual reality systems that consume such datasets.

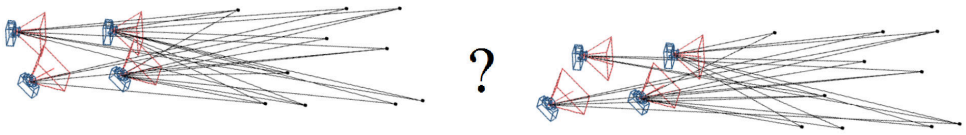
Two contributions to the field of photogrammetric computer vision are claimed, starting with the investigation of the theoretical accuracy potential of a set of imaging configurations comprising forward-facing cameras for dense reconstruction of narrow corridors using a network design approach suitable for cases where the geometry of a non-signalised scene cannot be accurately represented. In addition, it is shown how an interactive method for image segmentation can be automated, with the integration of geometric information, and used for the recognition of objects of interest, namely the track and its start line.

Finally, a sweeping-and-morphing algorithm for the extraction of cross-sectional curves of in-

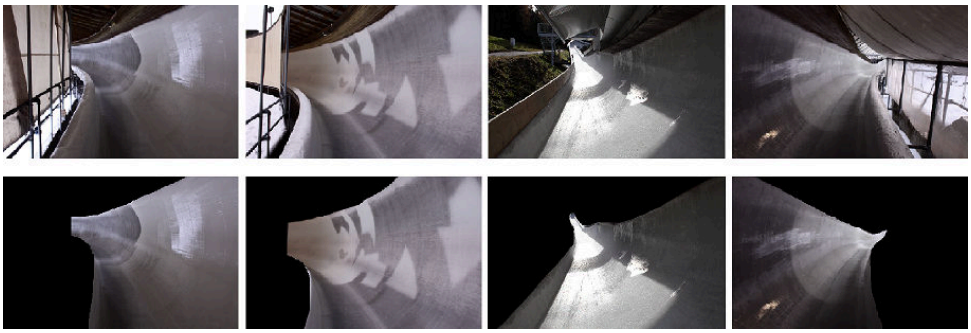
terest from a 3D triangle mesh with noise, gaps, and clutter in order to feed a lofting or path-based skinned surface approximation scheme, is considered to be the contribution to computational geometry and computer-aided geometric design. In contrast to existing approaches that deal with fitting simple geometric primitives—such as planes, spheres, or cylinders—to point clouds with low levels of gaps and noise in order to return watertight shapes, the proposed sweeping-and-morphing algorithm detects the boundaries of a free-form surface and is able to compensate for significant gaps along and across a centreline alignment. The surface fitting algorithm is complemented with an automatic clothoid spline fitting method for the horizontal alignment extraction of as-built tracks, which can also be applied to roadway and railway infrastructure for their as-built documentation. The clothoid spline fitting method improves on previously proposed techniques, which required user interaction or oversegmented the input, as it is fully automated and is tailored towards discovering the design intent of the alignment.

1.5 Thesis Structure

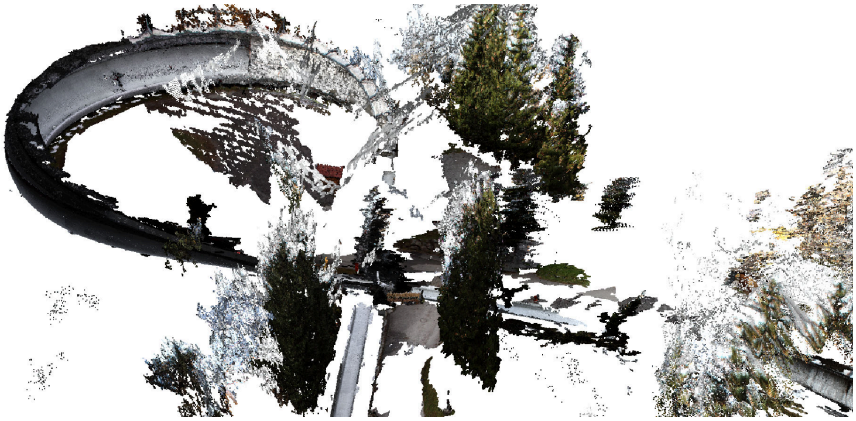
This thesis comprises five parts; the first part included an introduction to the subject as well as the problem statement, aim, scope, objectives, and motivation for the study. The literature review that follows, in part II, not only discusses the capabilities and shortcomings of existing techniques relevant to 3D scanning, reconstruction, and modelling of real scenes for the creation of virtual environments but is also complemented with a detailed description of existing sliding tracks, in chapter 2, and results from relevant preliminary investigations on track reconstruction, q.v. chapter 5. Chapter 6 synthesizes the literature review and provides an overview of the methodological approach that will be used towards achieving the objectives. In part III, the identified challenges are formulated as research questions and potential solutions are hypothesised. Guidelines for data acquisition, supporting the image-based 3D scanning of tracks, are provided in chapter 7. The results of experiments on curve and surface segmentation and fitting that were designed in order to gain specific answers with respect to the effectiveness of the proposed solutions, are presented in chapter 8 and form the basis of the proposed track scanning workflow which is described in detail in part IV. Two case studies are also presented therein, in order to verify the results in practice, within realistic environments. The fifth part discusses the findings and leads to the conclusions of the study. Additional explanatory information is included in the appendices.



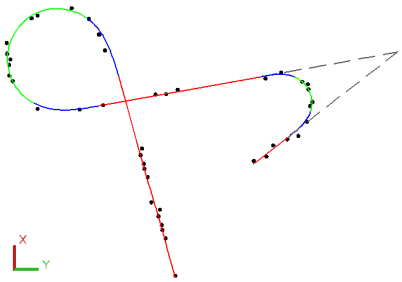
(a) Photogrammetric network design for a hand-held, forward-facing camera.



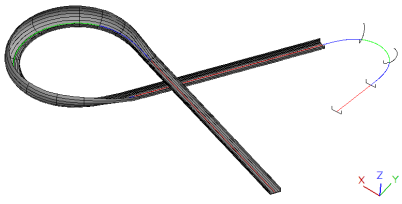
(b) Automatic image segmentation for track extraction.



(c) Track surface extraction from an incomplete, noisy point cloud of the site.



(d) Clothoid spline fitting for horizontal centreline alignment extraction.



(e) A sweeping-and-morphing process that will feed a lofting or path-based skinned surface appoximation algorithm.

Figure 1.2: Schematic illustration of the main objectives within the automatic track reconstruction workflow.

Part II

Status Quo

This part presents existing knowledge on the topic of generating a virtual environment for a sliding sports training system, interactive or not. The literature review is focused on the first two steps of the virtual environment creation pipeline as described in McGlone (2013, pp. 1045–1046), q.v. figure II.1; namely data capture and geometric reconstruction, q.v. chapters 3 and 4, respectively. Aspects of texture mapping and rendering are only briefly discussed where necessary as, on the one hand, the specific technical requirements for these tasks are primarily defined with respect to the resources available for the last step of the implementation, namely the system's interface and interaction. On the other hand, due to the nature of sliding sports and the high speed at which the interaction takes place, the visual fidelity is not directly correlated with high resolution, as is normally the case. Nevertheless, the inclusion of objects around the track with a realistic appearance at their actual position can make a significant difference between an as-designed and the as-built track model.

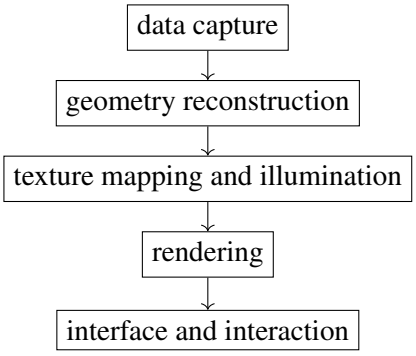


Figure II.1: Virtual environment creation pipeline; adapted from (McGlone, 2013, p. 1046,fig. 13.1)

There is a trade-off to be made on every quality-, effort-, and cost-related parameter when choosing a method for 3D reconstruction of real-world scenes. No single surveying method will meet all the requirements for any given project—especially when specific restrictions are imposed with respect to time, automation, and site accessibility—therefore, a combination of data sources is usually the answer to most problems. This work is concerned with image-based 3D reconstruction but the performance of terrestrial and hand-held laser scanners are also discussed in sections 2.3 and 3.6.2, respectively. Two preliminary trials with image-based techniques are presented in chapter 5, so that the reader becomes familiar with the object of interest and aware of the challenges involved.

Chapter 2

Virtual Environments in Sliding Sports Training

virtual reality systems

Apparently there is no consensus on the definition of virtual reality (Sherman et al., 2002, p. 6; Brey, 2008, p. 326; Hale et al., 2014, p. 23), which can be described as the perceptual illusion of presence in an environment different from one's physical surroundings without acknowledging the mediation of technology. Virtual reality systems require a virtual environment and the sensation of being "inside", while being able to sense and interact with it (Steuer, 1992; Biocca et al., 1995; Lombard et al., 1997). Systems that exhibit a vivid representation of reality and real-time interaction with the user (Steuer, 1992, p. 11) have been proven to be effective in skills acquisition (Lombard et al., 1997, p. 22).

There is a distinction to be made between virtual reality and computer simulation systems in that the latter primarily use mathematical abstractions of the physical systems they are describing without requiring realistic visual or human-machine interaction models (Brey, 2008, pp. 363–364). Nonetheless, acquiring and representing the environment's geometry is essential when modelling how sports equipment interact with it. In addition, when modelling the interaction between an athlete and the environment, its optical characteristics are necessary as well. Sliding sports can benefit from both system architectures in a number of ways, including training and sled design optimisation, under the premise that modelling approximations and uncertainties are kept to a minimum and the virtual environments are of high fidelity.

virtual environment

The digital representation of reality is where computer vision meets computer graphics; images of a scene are used in order to create new images of the same scene from different viewpoints. This can be accomplished by image-based rendering techniques without explicitly estimating the scene's geometry (Shum et al., 2008). However, the feedback that is required by most virtual rea-

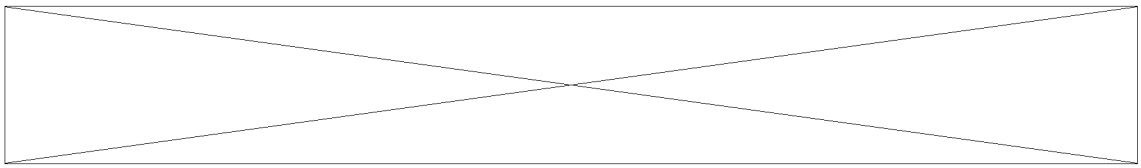


Figure 2.1: Forces acting on a sled during a run down the track; adapted from (RAENG, 2018). Copyrighted material removed.

lity systems does not only include the human visual system but also the haptic, proprioceptive, and vestibular systems systems that sense motion, balance, resistance, and pressure (Sherman et al., 2002); therefore, accurate geometric information is essential. Nevertheless, even though visual and geometric content based on real-world—rather than synthetic—data is more desirable (El-Hakim, 2004, p. 1015), reality is neither realistic nor irreplaceable as a fidelity objective (Northam, 2000).

The first step towards designing a non-intrusive surveying workflow for the generation of a virtual sliding environment was to get familiar with the nature of the sport and its natural habitat; namely, the track. The act of sliding down a track on a sled, even though technically challenging to master, is quite straightforward, as there is only so much one can do against gravity, friction, and drag, q.v. figure 2.1—for more details on the sports themselves see IBSF (2017). The governing body for the sliding sports is the International Bobsleigh and Skeleton Federation (IBSF), formerly known as the Fédération Internationale de Bobsleigh et de Tobogganing (FIBT). The majority of the rules, which are shared between the three sports, apart from defining the competition procedures, are concerned with enforcing the minimum safety precautions, setting specifications on sled design, and imposing restrictions on equipment that are considered to provide unfair competitive advantages. For example, using electronic equipment on board the sleds or wearable sensors on the athletes during official races and training sessions is prohibited; video recordings are restricted as well (IBSF, 2016, p. 26). There was an exception in the Sochi 2014 Winter Olympics for bobsleighs to carry sensors that can transmit their speed and acceleration in real time (OMEGA, 2014), which may be widely adopted in the future. However, as such data are being made available to all competitors, added value can only be created by the intelligence each team can produce out of it.

the sport

sports rules

Naturally, the sport is extremely competitive and most aspects are secretive (Bromley, 1999, p. 13). The sleds are covered when not on the ice and not much information can be acquired or knowledge exchange can happen by watching a sled moving at a speed of 125 kilometres per hour. Thus, aspiring champions primarily rely on skills and knowledge that are acquired through years of training (Levy et al., 2007). Most countries that are seriously involved in the sport, and have the necessary resources, invest on technologies that will give an advantage to their national teams;

competition

some of them, unfortunately, go a bit further than that.[1]

In the next couple of sections, general characteristics and design details of tracks are presented, q.v. section 2.1. The way their geometry has been represented and used in computer simulations so far is discussed in section 2.2.2. It is worth noting that there can be quite a few problems in gathering all the necessary information as the design and as-built plans of each track are not made available to the public by IBSF. Especially for older tracks, such information might not even be available or exist in a form that cannot be of use, q.v. section 2.3. Lastly, not every detail of each virtual system is likely to be disclosed, as each team wants to maintain a competitive advantage.

2.1 Combined Bobsleigh, Skeleton, and Luge Tracks

official competition tracks

The general geometric characteristics of combined bobsleigh, skeleton, and luge tracks which are currently being used in official competitions organised by the IBSF are outlined in table 2.1; schematic diagrams of their horizontal geometry are depicted in figure 2.2. Even though these values were extracted from IBSF's website and the official Olympic Winter Games reports, they are presented here for reference only; not only are they not sufficient to reconstruct a track model but several discrepancies were also identified between the reported values and those on the corresponding official websites for each track. Firstly, there is no consistency in the provided data type, e.g. height differences refer to either the track's start and finish lines or highest and lowest locations. Secondly, the track lengths may be ambiguous, as different start or finish lines exist for men, women, and each sport. Lastly, changes to the track might have been made since their construction or use during the Olympics, which may entail minor profile modifications—under the supervision of the IBSF—to entire sections being rebuilt (Altenberg, 2016). In addition, they do not comprise a single rigid structure but can consist of several segments—as they might have different needs in terms of temperature regulation (XVII Olympic Winter Games, 1994, p37)—which means that they might be variably affected from erosion. In fact, some tracks have adjustable supports (XVI Olympic Winter Games Organising Committee, 1992, p. 93) and, in any case, allow for minor movements due to temperature changes (SAIT, 2012, p. 89).

track design

The design and construction of an artificial[2] bobsleigh, skeleton, and luge track can cost up to $30 million (XVII Olympic Winter Games, 1994, p. 37; Sabbioni et al., 2016, p. 49) and

[1] In light of the publication of the second report investigating the allegations that Russian athletes have benefited during the 2014 Winter Olympic Games in Sochi from a systematic, centralized, and institutionalized manipulation and cover up of the doping control processes used in Russia (McLaren, 2016).

[2] The Olympia Bobrun St. Moritz-Celerina is the only natural track in the world; it is being built from snow annually.

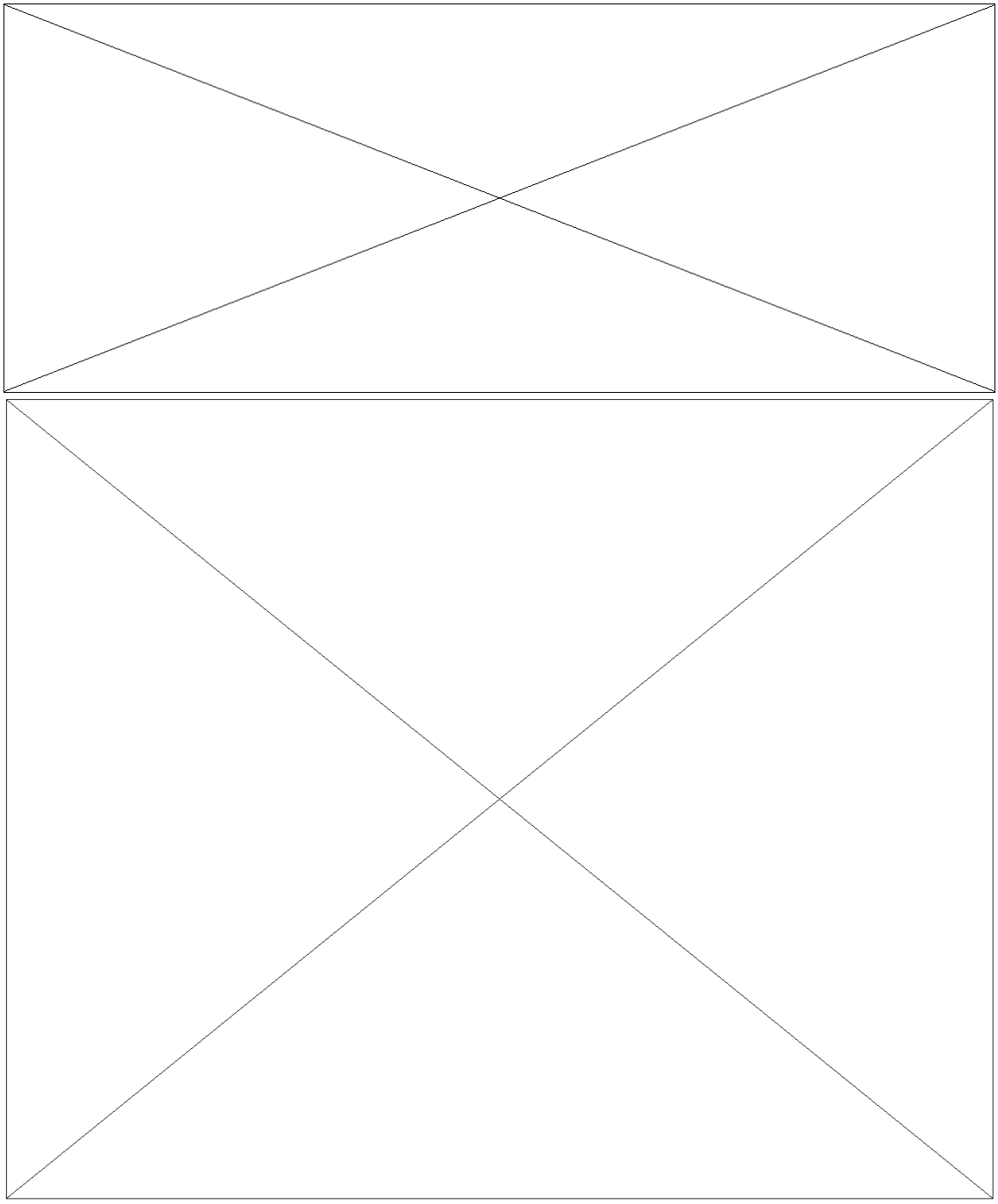


Figure 2.2: International competition hosting tracks, reproduced from IBSF (2017); see table 2.1 for details. Copyrighted material removed.

Table 2.1: Geometric properties of combined bobsleigh, skeleton, and luge tracks.

Track		Length (m)		Altitude (m)		Gradient (%)		Vertical	Number
Id	Name	Total	Competition	Start	Finish	Max	Average	Drop (m)	of Bends
1	Altenberg	1413	1413	785	660	15	8.66	122.22	17
2	Calgary	1494	1494	1251	1130	15	8	121	19
3	Igls	1478	1207	1124	1006	18	9	124[a]	14+1
4	Konigssee	1675.4	1251.2	730	630	-	9	120	16
5	La Plagne	1707.5	1507.5	1684	1559	14	8	124	19
6	Lake Placid	1680	1455	-	-	20	9	128	20
7	Lillehammer	1710	1365	-	-	15	8	114	16
8	Nagano	1700	1360	1028	922	15	7	112	15
9	Oberhof	1354.5	1069.7	830	735	36.4	9.2	95	15
10	Park City	1570	1335	2232	2128	15	8	104	15
11	Pyeongchang	1659	1376	930	850	25	9.48	117	16
12	Sigulda	1420	1200	117	18	9	8	99	16
13	Sochi	1814	1500	836	704	22	20	124	17
14	St. Moritz[b]	1962	1722	1852	1738	15	8	130	19
15	Whistler	1700	1450.4	935	802	20	9	148	16
16	Winterberg	1609	1330	760	665	15	9	110	15

[a] This is the value from the official IBSF (2017) website.

An unofficial Wikipedia (2018) list gives 98.1 m and 124.8 m (the latter is presumed to be the vertical drop of the old track, which has not been used since 1973).

The track's official website (Olympia, 2018) defines the vertical drop as approximately equal to 100 m; this value is used as an initial approximation in section 11.3.

[b] The only natural track in the world.

requires a high level of engineering expertise. There are very few engineering consultancy firms in the world that design tracks (SAIT, 2012, p. 37) but it might come as a surprise that the tracks built for the last seven Olympics were designed by one and the same engineering firm, namely IBG+PARTNER (2017). It follows that specific guidelines on the design of their geometry, apart from a generic outline of the design philosophy (SAIT, 2012, p. 38) and those that are used in road design pertaining to accommodating the dynamic movement of vehicles, are naturally considered secrets of the trade and thus not publicly available. To the best of our knowledge, a track design manual has never been published; it would probably not make sense for such a guide to even exist.

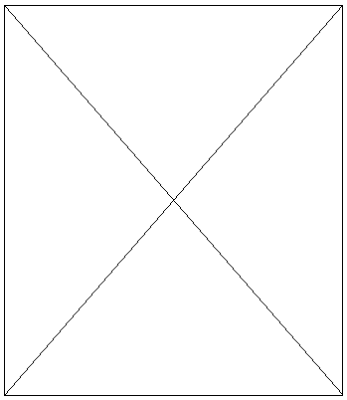
A small set of guidelines used to be part of IBSF's rule book and have been applied to tracks constructed until the last Winter Olympics, q.v. FIBT (2013) cf. IBSF (2016). The design intent of the tracks and the desired characteristics of a safe sled-run are described to an extent therein, without explicitly defining their geometric elements in detail. Traditional approaches which relied on hard-coded thresholds on the dimensions of geometric entities could be applied towards the design of controlled environments, i.e. a roadway. However, invariably applying them to any track site was soon proved to result in tracks that where either too conservative or, as the athletes became more skilled and the equipment more sophisticated, too fast; thus, more dangerous. Until the mid nineteen-eighties, tracks were constructed using the trial-and-error method but since then they are designed in an iterative fashion, based on general design principles and numerical analysis, so that only minor changes due to unanticipated factors are required during construction (Hubbard et al., 1989, p. 223; Stoye, 1990; SAIT, 2012, p. 4).

design methodology

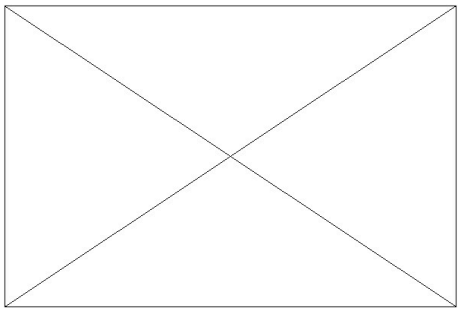
The study of such 3D shapes can be simplified using the classical method of slicing, which reduces them to a set of planar curves (Koenderink, 1990): a horizontal alignment, a vertical profile, and a set of cross sections with smooth transitions that has to accommodate all possible trajectory paths of bobsleighs, skeletons, and luges. An initial design of the track's centreline, with its vertical profile roughly following the site's surface relief, can be based on the designer's experience and general guidelines with respect to the maximum and average gradient. This is iteratively adjusted until the maximum level and duration of acceleration and speed throughout the track's length are within acceptable limits (FIBT, 2013). Tracks are constructed on the north side of a hillslope with the consideration of environmental protection and optimal use throughout their lifetime. Existing tracks are approximately 1400 to 1800 metres long comprising 14 to 20 curves. The competition length is a couple of hundred metres shorter to accommodate a deceleration stretch with gradient up to 12%—or 20%; another inconsistency in FIBT (2013, pp.39,41). The horizontal alignment is represented by a clothoid spline that comprises straight line segments

geometry

alignment



(a) A sketch of the timing system that was installed at the Sochi track; adapted from The Swatch Group Ltd (2014).



(b) The timing system at the start line of the Sochi track; reproduced from The Swatch Group Ltd (2014).



(c) The push-off stretch at the Lillehammer track.



(d) The starting straight at the Igls track.

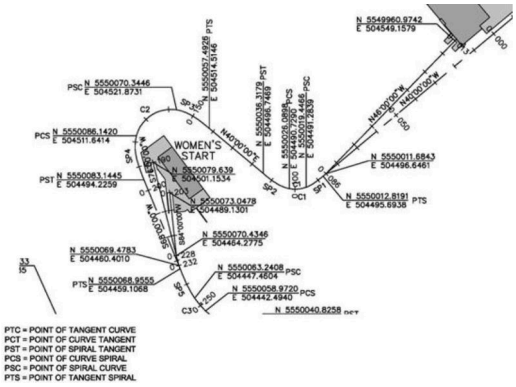
Figure 2.3: The typical starting area and timing system of a sliding track. Copyrighted material removed.

and circular arcs connected by clothoid transitions (Mossner et al., 2011, p. 892;SAIT, 2012, p. 38;Rempfler, 2015, p. 52), q.v. figure 2.4a. The minimum arc radius in the middle of each curve can be as small as 10 m and the parameters of the in-curve and out-curve clothoid transitions at each corner need not be equal (Mossner et al., 2011, p. 893). It might also include bends with turning angles more than 270°, e.g. the Igls, Konigssee and Calgary tracks, called *kreisel*, which is the German word for the roundabout or the spinning top toy. The alignment's vertical profile curve is usually modelled using straight segments joined with parabolas (Kuhn, 2013). The 75 m long starting straight comprises a 15 m long push-off stretch with a gradient of -2% leading from a wooden block to the approximately 3 m wide start line; for the next 60 m, where the width is gradually reduced so that in the iced state at the straights is less than 1.4 m, the gradient should be -12%, q.v. figure 2.3.

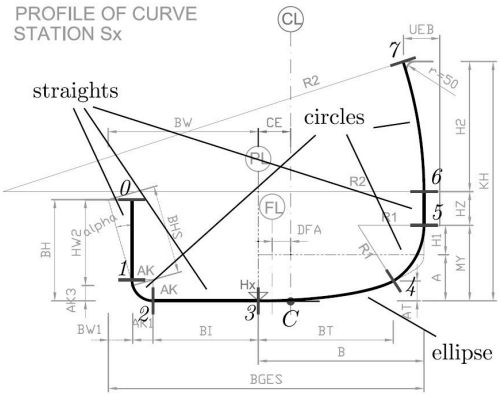
profile

The rectangular cross section at the straights has a nearly vertical wall on either side, which must be at least 0.5 m high at the iced state, q.v. figure 2.4d. A fillet with a 10 cm radius in the

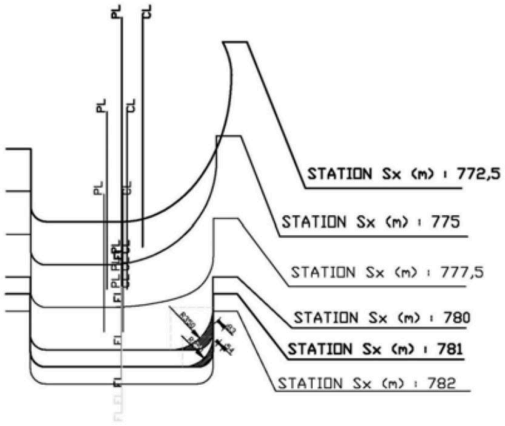
cross section



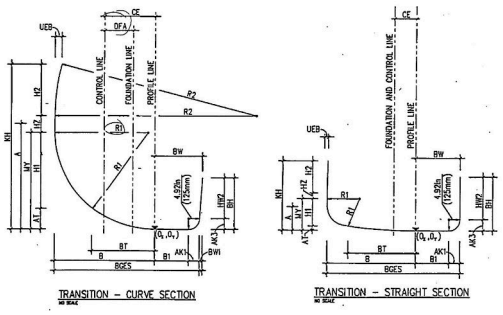
(a) Horizontal alignment, reproduced from SAIT (2012, fig. 4.1, p. 88)



(b) Typical cross section curve segments, reproduced from Rempfler et al. (2016)



(c) Stacked cross section reproduced from SAIT (2012, fig.4.14, p. 109)



(d) Bend and straight segment cross sections, reproduced from Hubbard (2013)

Figure 2.4: Whistler track geometry.

iced state is used for the transition from the flat floor to the walls. The fillet on the base of the outside wall at a corner allows a smooth transition from the straight segment. However, the inside fillet, present in most track designs, is considered to be a design flaw (Hubbard, 2013). Along the corners, as well as along the transition segments between them and the straights, the inner half-section retains its shape except for the variable half-width of the flat floor. The outer half-section comprises a combination of straight, circular, and elliptical segments leading to a wooden guardrail nearly perpendicular to the tracks surface to prevent the sled from exiting the track, q.v. figure 2.5.

Whereas, normally, the mathematical description of the horizontal alignment and vertical profile of a road refers to the same 3D line, in this case it does not. In figures 2.4b, 2.4c, and 2.4d, the projection of the track's centreline and profile line on the cross sections are indicated by the



(a) Scrapping the ice track. **(b)** Scrape marks. **(c)** Spraying water on the track.

Figure 2.5: Track surface preparation: Embedded in the concrete structure is a cooling system comprising up to 80 km of pipes, which brings the track's temperature to well below $-15°C$ by circulating liquid ammonia or other freezing agents (XVIII Olympic Winter Games, 1998, p. 228). Water is sprayed on its surface so that a layer of ice—approximately 1 to 6 cm thick that can go up to 10 cm towards the end of the season—that follows the cross-track profile of the concrete surface is created and maintained at around $-10°C$ (XVI Olympic Winter Games Organising Committee, 1992, p. 93; Altenberg, 2016). The ice surface is manually and regularly swept (FIBT, 2013, p. 20) and scraped.

codes *CL* and *PL*, respectively. It is obvious that these design elements are theoretical and not only do they not coincide with but they are also not parallel to any physical structure, except along the straight segments. The track's design alignment, i.e. the projection of the horizontal clothoid spline to the track's surface, is a design element that is not defined unambiguously as it is only used in order to define an envelop that includes all possible sled trajectories. On the other hand, the vertical profile models the track floor, which is the lower flat part of the cross section. Detecting this line in a point cloud, in order to infer missing parts using the knowledge about its geometry, is very difficult, as it lays on a smooth surface and is covered with a smooth layer of ice, q.v. figure 2.5.

2.2 Winter Sports Simulation

Virtual reality systems can safely and effectively allow skills acquisition and remote knowledge transfer about real-world environments when physical presence is either unsafe or impractical (Garrett et al., 2010, pp. 43–44). Performing a potentially lethal activity (SAIT, 2012) in the safety of a virtual environment alone makes the case for the necessity and importance of sliding sports simulators. Sliding sports benefit from such systems in more than one way as they are not only useful to athletes and managers but also to track and equipment designers (Hubbard et al., 1989, p. 223; Mossner et al., 2011, p. 893; Kelly et al., 2000, p. 13; Sabbioni et al., 2016, p. 49).

Training is extremely important in any sport, as it allows the athlete to develop the necessary skills which are required in order to anticipate and react to any situation that may occur during competition. It also introduces new athletes to the sport, prepares injured athletes to return in action, and, more importantly, helps them to prevent such injuries from occurring in the first place. For most sports, accessing the specific site where the competition will take place is not that important, as training routines can take place on alternative sites that pose some of the characteristic properties of a track, field, or court. In contrast, in racing sports that involve non-canonical track designs, which cannot be replicated elsewhere, certain aspects of training fall short of the real experience. The fact that the few winter sports training facilities in the world are only available between October and March for training on ice, makes it even harder for athletes to gain access.

training

Traditionally, in an attempt to bring the track to the athlete with the help of technology, video recordings are used in sliding sports to monitor performance during training or competition and subsequently adjust it after the fact (Bideau et al., 2010, p. 16). These recordings can convey information about the track and the surrounding environment and they can do so from the athlete's point of view. However, this method is not as effective as one that combines physical and visual cues that correspond to the actual track and are essential in achieving an immersive experience with real time interaction (Lombard et al., 1997, p. 15). Sliding sports athletes need skills similar to those required by pilots, as they are faced with challenges of the same nature (Levy et al., 2007). Systems that can produce such experiences, e.g. flight simulators, have been proven quite effective for pilot training (Lombard et al., 1997, p. 22). However, for sliding sports, the performance cannot be significantly enhanced unless training sessions are conducted under the same conditions as in the actual race, which involves the athlete being subjected to acceleration up to five times the acceleration due to gravity and synchronising the necessary muscle memory with site specific visual cues (Levy et al., 2007; Craig, 2013, p. 163).

visual cues

Virtual reality systems are being used extensively in many sports that do not necessarily have similar problems of accessibility or safety risks. They can also be used for relatively simpler tasks that do not necessarily require real-time interaction such as strategy planning and analysis (Miles et al., 2012). One of the most important aspects in sliding is discovering optimal trajectories for the whole or a part of a track. This task is equally important in educating young athletes and preparing more experienced ones (Zhang et al., 1995; Rempfler et al., 2016). In addition, such systems can be used so that the geometric characteristics of a track can be optimally selected at the design stage and for the detection of design flaws after construction (Stoye, 1990; Braghin et al., 2010; Hubbard, 2013).

trajectory analysis

In sports that involve interaction between humans and bespoke equipment, computer simulations are a great way of controlling certain parameters, e.g. environmental conditions during trials, in order to avoid solely relying on the subjective athlete feedback from experiments towards improvements and innovations. This is particularly important in skeleton or luge where on-board sensors might affect the athlete's performance and not many trials are possible due to track availability and athlete fatigue (Braghin et al., 2011; Sabbioni et al., 2016).

equipment optimisation

All the above tasks can only be reliably performed if both the 3D and mathematical models which describe the track geometry and the physical processes that occur during the slide, respectively, are representative of the actual ones. In the following sections, the approaches that have been used in some of the existing systems with respect to the aforementioned models are reviewed; in particular, how the geometry is created, represented, and used in the simulating systems and the level of abstraction and sophistication of the physics models.

2.2.1 Track-Sled-Driver Interaction

There are three essential modules in a dynamic system which simulates a bobsleigh, skeleton, or luge run: a track model, a sled model, and a driver model. In addition, a model for the interaction between the sled and the track, which describes the sled's motion under the influence of all the forces acting on it including the driver's steering, is required. All of the above can vary in complexity and sophistication depending on the simulation purpose and the available resources (Sabbioni et al., 2016, pp. 225-252).

dynamic modelling

The simplest configuration is that of a sled represented by a point mass, known as a particle model, moving along a specific fixed trajectory that is assumed to be optimal in some sense—e.g. fastest, safest, or easiest—or is under investigation during a track design process as a potentially optimal one. The 3D position of the sled can be described by a single degree of freedom, i.e. the abscissa s along the curve that represents the trajectory, which is in turn a function of time $s(t)$. No driver model is required in this case and, in theory, not even the track surface as long as the forces acting on the sled at each position along the trajectory are known; either by direct measurements or as initial approximations within a track design optimisation process. The motion of a sled on the track is governed by three external forces acting on the point mass due to gravity, aerodynamic drag and lift, and its contact with the track surface. Assuming a rigid track surface, the latter can be analysed in three components: one normal to the track surface at the point of contact and two more, due to friction along the direction motion and due to the resistance to lateral skidding, lying on the tangent plane.

particle model

Such simple models are often used for sled design or sport training purposes as they are adequate for the analysis of the effects of the sled's mass, aerodynamics, and friction under different starting conditions representing the sprint phase. By adding one more degree of freedom to the track model, the trajectory of the sled can vary. If the geometry of the track's surface is known, the optimal trajectory can be discovered using the aforementioned simple single particle model or by introducing a more complicated rigid or full body model of the sled. Moreover, a driver model can be used to take into account the steering forces due to the athlete's response, which is necessary to control the sled and guide it along the desired path.

2 DoF

The sled and driver models are only mentioned here to point out the fact that various interactions can occur between them and the track. The sled runners may have multiple contacts with the ice surface—or lose it altogether because of a bump—and the driver's feet may interact with the track's floor or the sidewall. The more complicated the system, the more essential a reliable and realistic 3D model of the track becomes for design and verification of tracks and sleds. How these as-designed models can be and have been created and represented in such systems is presented in section 2.2.2.

track model

2.2.2 Mathematical Representation of the Track's Surface

The implementation of a virtual reality system for sliding sports requires adequate mathematical representations of the models that are used in each individual module, q.v. section 2.2.1. In order to describe the sled's motion, its position at any time on the track and all the possible positions on the track's surface need to be expressed in a common framework. This can be efficiently achieved by reducing the 3D surface of the track to a 2D parameter space (Botsch et al., 2010) so that the coordinates of any position on it can be expressed in the form

parametric form

$$\boldsymbol{r}(\alpha,\beta) = [x(\alpha,\beta), y(\alpha,\beta), z(\alpha,\beta)]^T \tag{2.1}$$

such that its full extents are covered by an $m \times n$ grid, q.v. figure 2.6.

The sled's position is also expressed with respect to the same dimensionless parameters α and β, called the generalised coordinates (Kane et al., 1985, p. 37), which are in turn a function of time t. Considering an initial state of the sled and all the forces F^i that act on it, within the simplest case of a particle model with two degrees of freedom, its generalised speed u and position on the track $\boldsymbol{r}(\alpha(t),\beta(t))$ at any time t can be estimated by a set of four first order, ordinary, nonlinear differential equations (2.2) to (2.5) (Hubbard et al., 1989). These are schematically presented below, as the exact motion model is beyond the scope of this work, in order to demonstrate the

motion

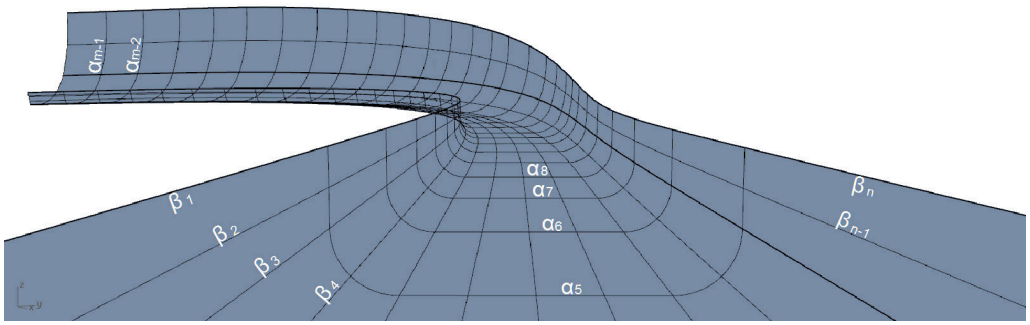


Figure 2.6: Track surface parameterisation.

fact that the calculations rely on the partial derivatives of the parametric surface representation of the track.

$$u_1 = \frac{\partial \alpha}{\partial t} \quad (2.2) \qquad \frac{\partial u_1}{\partial t} = \phi_1(F^i, u_1, \frac{\partial \boldsymbol{r}}{\partial \alpha}, \frac{\partial \boldsymbol{r}}{\partial \beta}, ...) \quad (2.3)$$

$$u_2 = \frac{\partial \beta}{\partial t} \quad (2.4) \qquad \frac{\partial u_2}{\partial t} = \phi_2(F^i, u_2, \frac{\partial \boldsymbol{r}}{\partial \alpha}, \frac{\partial \boldsymbol{r}}{\partial \beta}, ...) \quad (2.5)$$

The geometry of the track's surface is too complex to be described by a single, simple analytical expression. Hence, an appropriate abstraction for its digital representation is required. Computer generated models of imaginary tracks can be created in any convenient shape and form readily available for the motion estimation module. However, the geometry of real tracks, that as-designed is of concern herein, usually comes in the form of design and construction plans. These already provide a convenient implicit parameterisation in the form of planar cross-sectional curves extracted along a space curve, i.e. the alignment. Even if an analytical form for the alignment and cross-sectional curves' geometry is not provided, a parametric representation can be constructed by extracting points along each one of them and approximating their shape using an appropriate parameterisation in order to define the grid. The functional surface model, e.g. a polynomial, with respect to the parameters α and β for each rectangular patch on the grid is selected so that the corresponding 3D coordinates of any position on the track can be interpolated and its derivatives analytically calculated.

Even though a mathematical artifact such as the curvature discontinuity of an ellipse or an arc tangential to a straight line which represents the track's floor may not significantly affect the visualisation of the track's surface, the surface model at each patch should ideally pass through the grid points and have a certain degree of continuity at the edges in order to reflect the smoothness of the sled's motion on the ice. This is of course an implementation choice and, together with the resolution of the grid, depends on the available resources and is a matter of balance between

economy and performance.

parameterisation

All existing system implementations use some variation of the aforementioned scheme at varying resolutions. Hubbard et al. (1989) investigated bicubic splines and B-splines for the patch surface interpolation on the Calgary track using $n = 10$ for the point extraction from as-designed cross sections. Zhang et al. (1995) used B-spline interpolation between "realistic track surface shapes" along an alignment of the Lillehammer track with $m = 800$ and $n = 21$ which correspond to approximately 2 m along and 0.1–0.3 m across track, q.v. figure 10.8. Huffman et al. (1996) used bicubic spline interpolation for Calgary, La Plagne, Lillehammer, and Salt Lake City and reported subcentimetre interpolating errors with respect to the as-designed geometry. Kelly et al. (2000) also used bicubic spline to interpolate the construction plans for Salt Lake City using $m = 650$, which correspond to approximately 2 m along the track. The simulator that was created by the University of Calgary was based on the Salt Lake track digitisation from data available in the public domain (Levy et al., 2007). Roche et al. (2008) used satellite imagery to extract the St. Moritz track axis and track plans in a motion model with a single degree of freedom; cubic spline interpolation was used in this study. Braghin et al. (2010, 2011) used as designed plans and some on-track measurements with $n = 60$ and $m = 3000$ for the Cesana track and $m = 725$ for Whistler. Instead of interpolating between 4 grid points they fit a paraboloid to 11 of them. In Mossner et al. (2011) the as-designed plans of Whistler were used with a single degree of freedom model. Gong et al. (2016) estimated the cross sections by taking measurements on site—the nature and means of the measurements were not described—and extracting the alignment for the Igls track from satellite imagery creating an 890×14 grid that corresponds to approximately 1.5 m along track and 0.1–0.3 m across track. Rempfler et al. (2016) reconstructed a NURBS surface of the Whistler track from the construction plans. The same plans, with $n = 100$ and $m = 3034$, were compared to as-is concrete and ice surface data in the Whistler Track Construction Study (SAIT, 2012), q.v. section 2.3.

2.3 Whistler Track Construction Study

A technical study of the as-built Whistler track was conducted between 03/2011-10/2012 by the Southern Alberta Institute of Technology (SAIT) as part of the investigation of the death of the Georgian luge athlete Nodar Kumaritashvili during the 2010 Winter Olympics in Vancouver Canada. This study was the first of its kind (SAIT, 2012, p. 2) and, to the best of our knowledge, the only one since. The aim of the topographic survey that was conducted as part of that study was to assess if the concrete structure of the track was constructed within the design specifications. In

addition, an attempt to determine the typical thickness of the ice layer along the track was made by measuring profiles at predefined marked locations.

topographic survey

The survey included the establishment of a survey control network and the laser scanning of an instance of the as-constructed concrete and ice track surface at a specific point in time. Each one of these tasks lasted almost a week and required unobstructed access to the site. Terrestrial laser scanning was considered the most precise and cost-effective technique; robotic total stations, digital levels, and GNSS receivers were also used in order to achieve subcentimetre accuracy for the survey control network. The coordinate differences for the common points between the newly established survey control and those from the original network, which was used during the design and construction phase, that were discovered on site were 3-50 mm (SAIT, 2012, pp. 71–72). Two laser scanners serving two discrete roles were used in the process of creating the as-constructed track surface. A short-range phase-shift scanner was used for scanning along the track. A long range time-of-flight scanner was used to scan the surrounding site in order to assist with the registration of the individual scans along the track, as directly referencing based on common targets between them was deemed impractical. The laser scanning report indicates that "paint was used in order to mark the ice surface and collect measurement at predefined profiles" because "The effect of varying degrees of translucency of ice has on the laser scan is not well known or studied" (SAIT, 2012, pp. 71–73,76).

as-constructed

as-designed

deviations

The design plans were used for the reconstruction of the 3D model of the as-designed track. There were over 2500 items related to the track design and construction; yet, this dataset appeared to be incomplete and included conflicting information. No as-built plans were available and no tolerances were specified for the conformance of the shape and position of the constructed track to the design intent, apart from generic shotcrete application tolerances (SAIT, 2012, p. 46,107,119,134). The comparison between the as-designed and as-constructed concrete track surface revealed deviations up to 7.3 cm. However, for the estimation of this value, data that were considered to be inaccurate or at places where there where scanning problems, were not taken into account. In order to rule out the possibility of the coordinate system transformation being responsible for the discrepancies, a best-fit alignment between the as-designed and the as-constructed track models was performed which led to even larger maximum deviations, up to 17.8 cm; the results from this approach were not considered towards the conclusion of this study.

At some locations, the ice profile appeared to be below the as-constructed concrete profile but the magnitude of this discrepancy, which would be indicative of the precision of the laser scanning survey, was not defined. The deviations that were found between common points in the two scans,

which were considered stable, at the vicinity of these cross sections were found to be consistent with the overall subcentimetre registration accuracy between the concrete and the ice scan point clouds. Thus, the discrepancy was attributed to the limitations in the laser scanning process; more specifically to the registration errors between the individual scans (SAIT, 2012, pp. 42,45,129–133,137).

conclusion

The conclusion of the study was that

> Given the tolerances that can be achieved for this type of construction on the side of a mountain, recognizing that the comparisons may be influenced by the accuracy of the surveys, concrete shrinkage and temperature movements, and noting that the alignment and profile are relatively smooth along the as-constructed track length, in our opinion the as-constructed geometry of the track generally meets expectations. (SAIT, 2012, pp. 46,134)

In summary, even for a well-organised survey of great importance, such as that one, which was conducted under more favourable settings than those being considered in the present study, several problems with respect to data collection and geometric processing are to be expected and the results may not always be aligned to expectations. The concluding remark implies that discrepancies up to 20 m between the as-designed and as-built geometry are considered insignificant. It can be said that a week's worth of work by a highly-skilled surveying crew with all the necessary geodetic-grade equipment and an established survey control network cannot be expected to yield a result of average accuracy better than 10 cm.

Chapter Summary

This chapter provided a detailed description of the typical geometry of a track in order to highlight the fact that inferring the design intent in order to recreate a track without the design plans and without user interaction is very challenging. A traditional survey with terrestrial laser scanners requires a team of trained surveyors, ice paint, and almost a week of unobstructed access on the site in order to achieve subdecimetre accuracy. However, as the track geometry that is being used within existing simulating systems has been based on the design plans and assumed to be perfect, no sensitivity analysis exists with respect to the accuracy or resolution requirements. In any case, such requirements would be dependent on the simulating system's specifications and beyond the scope of this thesis which is focused on the potential of the image-based 3D-scanning workflow for track scanning.

Chapter 3

Photogrammetric Computer Vision

origin

The origins of photogrammetry can be traced back and attributed to Euclid's Optics, Aristotle's notion of the camera obscura, daVinci's bulls-eye lens, or Kircher's magic lantern (McGlone, 2013, pp. 1–59). Shortly after the invention of photography in the 1850s, Laussedat—who is considered to be the first photogrammetrist—and Meydenbauer were independently using photographic equipment to measure building façades for architectural documentation. Thus, it can be argued that the technique was initially developed for close-range applications—imaging distances relatively small, less than 300 m (Luhmann et al., 2014), compared to the camera stereo-base leading to significant perspective distortion (Triggs et al., 1999, p. 352)—which is the primary topic of this chapter, and was further developed for aerial platforms or even underwater use (Li et al., 1997).

etymology

Meydenbauer is also responsible, together with Kerstner, for the term photogrammetry itself (Albertz, 2007; Grimm, 2007) which is inaccurately explained in the "Manual of Photogrammetry" as

> ...derived from three Greek works, photos, meaning light, gramma meaning something drawn or written, and metron meaning to measure. The root words, therefore, originally signified measuring graphically by means of light (McGlone, 2004, p. 1; McGlone, 2013, p. 2).

The misinterpretation probably stems from the fact that a similar term has also been used in the past, namely photographometrie or photometrographie (Grimm, 2007). The next edition of the manual should state that the term photogrammetry is derived from three Greek words: *photo-* which is the genitive (possessive) singular case of the word *phos*—meaning light—*gramme′*—meaning line—,and *-metria′*—a suffix which indicates that the prefix is involved in a measurement process. The root words, therefore, literally mean measuring a ray of light, cf. (Baltsavias, 1999).

The inclusion of the term *line* in the definition makes sense in light of a fundamental assumption in photogrammetry; namely collinearity, q.v. equations 3.1–3.2. More details on the physical and mathematical models that form the basis of photogrammetric techniques and are relevant to the track scanning application are presented in sections 3.2 and 3.3.

terminology

The above-mentioned discussion is indicative of the greater need for a common vocabulary that has already been identified by the photogrammetric community; yet, it is far from being widely adopted, as both fields are based on well-founded theories but use different terminology and definitions for quite similar tasks, cf. Granshaw (2016a,b) and Fischer et al. (2013). Consequently, as various applications arise in different contexts, users' attempts to fill the gap in order to describe their workflows result to the invention of redundant technical jargon by combining existing terms that are essentially overlapping or even tautologies. In the earth sciences, for example, where photogrammetric techniques have been successfully applied in domain-specific problems for years, the recently coined terms "Structure-from-Motion Photogrammetry" (Westoby et al., 2012; Javernick et al., 2014; Micheletti et al., 2015) and "Topographic Structure-from-Motion" (Fonstad et al., 2013) are being used extensively in order to stress the fact that user-friendly algorithms, which solve the reconstruction problem without demanding manual measurements on artificial targets with known positions in the 3D object space, are now widely available. However, the structure from motion (SfM) pipeline has been around for a few decades now (Ullman, 1979; Fitzgibbon et al., 1998), even though only recently efficiently implemented (Snavely et al., 2007; Davison et al., 2007) mainly due to the development of algorithms for robust estimation (Fischler et al., 1981) combined with sparse matrix techniques (Szeliski et al., 1994) for the bundle adjustment (Triggs et al., 1999) of feature point correspondences (Lowe, 2004).

Presumably, the new terms are used in an attempt to describe the notion of automating certain digital processes which, nonetheless, are not necessarily new. Perhaps they are new to a wide user-base but they are still photogrammetric methods so they should not be used to redefine photogrammetry.[1] In any case, once the feature points are detected and matched across all images using "yet another heuristic Structure from Motion (SFM) iteration" (Triggs et al., 1999, p. 299) the workflow "can revert to traditional rigorous photogrammetric processes" (Luhmann et al., 2016, p. 42). Instead of introducing fancy old terms for mature methods every time a new algorithm successfully implements them, the reader is referred to a definition according to Förstner et al. (2016) under which this emerging family of photogrammetric methods can gracefully fall under:

definition

[1] Surprisingly, this was precisely the subject of the editorial that was published in The Photogrammetric Record journal (Granshaw, 2018), on the day the manuscript of this thesis was finalised.

> Photogrammetric computer vision comprises photogrammetric theories, methods and techniques related to computer vision problems and relevant for automatically solving tasks in mapping and metrology using software systems and the necessary tools for design and evaluation of the results. As such it is intimately linked with methods from mathematics, statistics, physics, and computer science. It is closely coupled especially to methods in computational geometry, image processing and computer graphics.

In this chapter, the essential tasks for a successful terrestrial photogrammetric survey are outlined. Its success can be defined with respect to the accuracy and completeness of the result and the level of automation that was achieved. The available resources can be a limiting factor, so careful planning is required. The process of project planning, q.v. section 3.2, considers all factors that need to be taken into account in order to set out an initial parameter set that can be further optimised using the concept of photogrammetric network design, q.v. section 3.4. Certain properties of the scene or object of interest usually dictate some of the decisions that need to be made with respect to the selection of the camera characteristics and the image acquisition strategy which may potentially affect the quality of the outcome. The network design process is introduced after describing the bundle adjustment method upon which it relies. This method carries the computational effort involved in the camera orientation and object point coordinate estimation and can also yield quality metrics useful for the precision assessment of the reconstruction, q.v. section 3.3.1. Finally, a hand-held sensor that can provide direct measurements, which can either facilitate the photogrammetric workflow or can be used as a benchmark against which to compare its results, is briefly presented in section 3.6.2.

3.1 The Photogrammetric Process

image-based modelling

Image-based 3D modelling (Quan, 2010; Remondino et al., 2006) is not an easy task. Nevertheless, any multiple view analysis process, even the most complicated cases, can be described by one simple statement similar to the one that expresses the aim of this study, cf. section 1.2. The first step towards dealing with such problems—which, more often than not, cannot be modelled by a single analytical expression—is to translate them from plain English to a symbolic representation that allows further decomposition. Subsequently, a step-by-step solution can be devised by putting together well-defined methods which can be finely tuned and deal with noise and outliers using appropriate heuristics (Förstner et al., 2016, p. 9). Even though, under reasonable assumpti-

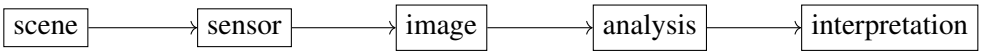


Figure 3.1: A metamodel for automated photogrammetric computer vision systems (Förstner, 1993; Förstner et al., 2016)

ons and sufficiently accurate initial values, precise estimations of the parameters can be expected, the integration of built-in self-diagnostics is equally important; especially, for automated usage by non-experts.

A metamodel for automated photogrammetric computer vision systems, proposed by Förstner (1993); Förstner et al. (2016), is depicted in figure 3.1. It represents the full processing pipeline for image-based 3D modelling from scene description to its 3D digital representation through the analysis of data collected with an appropriate sampling process. Essentially, it serves the purpose of conveying answers to questions like: metamodel

- Which part of the scene is of interest?
- How will the sampling be made?
- Which method will be used to process the raw data?
- Is there a method-specific data encoding scheme that has to be followed?
- What is the best way to represent the result?
- How can its validity be verified?

In this context, the data samples, which are collected by means of optical sensors and are assumed to sufficiently describe the geometric properties of the objects of interest, are subsequently transformed back to a 3D model. This is an inverse problem and, similarly to most such processes during which some information is lost, an ill-posed one (Kabanikhin, 2008). Getting a single solution requires the explicit integration of prior information that reflects existing knowledge about the scene or imposes specific expectations on it; the selection of a particular mathematical model to describe the physical processes involved can also implicitly affect the outcome.

This metamodel comprises two major components, namely analysis and interpretation. The analysis component is concerned with parameter estimation methods in the presence of uncertainty that can be divided in two broad areas. Analysis of geometry, which entails the tasks of project planning, camera calibration, image orientation, and object point determination and examines them in the unifying framework of the bundle adjustment, q.v. section 3.3. The automation analysis

that is necessary in some or all of the aforementioned tasks requires robust methods with cognitive elements for scene understanding such as feature matching, image segmentation, and texture classification, q.v. section 3.5. The ultimate goal of the reconstruction of a scene, which can be considered as a multi-stage transformation process, is its interpretation. It is fulfilled by the digital representation of its geometry that, depending on the input or output of each stage of the reconstruction, is not necessarily consistent throughout the entire workflow (Förstner et al., 2016, p. 442). A final transformation might be required in order to create the appropriate representation for a specific application, q.v. section 2.2.2. On the one hand, the result of the photogrammetric bundle adjustment is usually the 3D coordinates of a selected subset of point samples that can be densified in a subsequent step or accommodated by the parameters of geometric primitives, such as lines, planes, quadrics etc., which may be assumed to be present in the scene. On the other hand, active sensors yield a 3D point cloud without the complexity of an intermediate analysis stage. In order to maintain a unifying approach, a generic unstructured point cloud is considered to be the primary input to the interpretation component, which is covered in chapters 4 and 8.

interpretation

3.2 Project Planning

Project planning sets the ground so that certain, often counteracting, project targets related to accuracy, reliability, economy, time, and effort can be achieved. It provides tentative answers to the questions implied by the metamodel that was outlined in figure 3.1, by describing the nature of the object of interest, image acquisition and processing systems, supplementary observations, output format, verification process, and the required resources. As the number of combinations of possible choices for the values of the defining factors—which can vary for a different instance of the same problem—are infinite, an investigation of the nature of their interaction and effect on the target values of interest is essential.

An overview of the subject is presented in Luhmann et al. (2014, pp. 543–544), where the most important criteria for the initial planning phase are specified as the following: sensor type, object dimensions, accuracy and resolution requirements, environmental conditions, object targeting, supplementary measurements, measurement mode, and the available time on site and for post-processing. The sensor model of a perspective frame camera, which is commonly used to describe the imaging process for off-the-shelf point-and-shoot or digital single-lens reflex cameras (DSLR), is considered in this study. The imaging process can be adequately described, for most frame cameras used in photogrammetric applications, by an idealisation based on a thick lens model, q.v. figures 3.2 and 3.5. More details on the ideal central (single-viewpoint) projection, which

planning criteria

sensor model

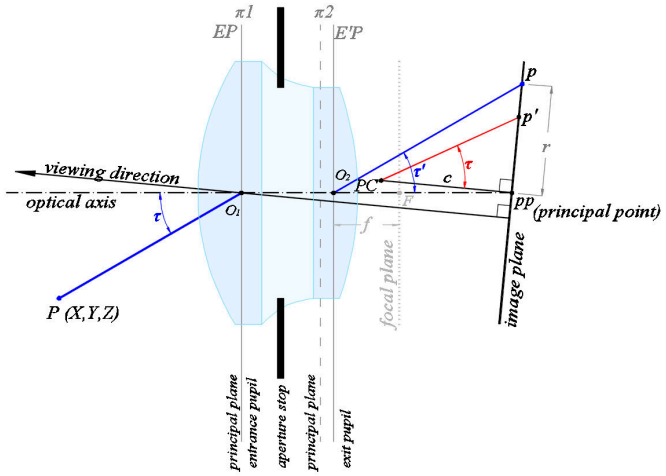


Figure 3.2: Thick lens model; adapted from Kraus (2007, p. 48–49;fig.3.1.1–3.1.3), Luhmann et al. (2014, p. 114,116,146; fig. 3.7-3.9,3.44-3.45), Förstner et al. (2016, p. 461; fig. 12.4).

is only in theory straight line-preserving, and the more sophisticated sensor model that is required for modern camera systems which are built with asymmetrical compound lenses can be found in Kraus (2007); Luhmann et al. (2014); Förstner et al. (2016). The remaining aspects pertaining to photogrammetric project planning are discussed, with respect to track scanning, in sections 3.2.1–3.2.3.

3.2.1 Site Constraints

object description

Details on the geometry of sliding tracks and sport related constraints were presented in chapter 2 together with a description of the desired output; i.e. a parametric model of the entire track surface, which will be used to estimate the sled's trajectory, and a representation of its immediate environment. Their combination will be used to produce 3D graphics within a virtual environment which will provide visual cues to the athlete. For the sake of simplicity, for survey planning purposes, it suffices to think of a track as 3 m wide and approximately 1.5 km long, partly covered, corridor. The accuracy requirements that were set by the key stakeholders for this track scanning project was a couple of metres for the overall track shape and a few centimetres for the across track profiles. This rather loose specification was based on the estimate of the expected deviations between the as-designed concrete and the as-built ice track. Given the conclusion presented in section 2.3 the latter is quite optimistic, yet achievable if only a segment of the track is considered. For example, optimising the trajectory along a single bend might be required. According to the assessment of simulated bobsleigh runs presented in Zhang et al. (1995, p. 18), an improvement in any individual bend can improve the overall performance. However, if the overall shape or

accuracy specification

orientation of the track is inaccurate, the initial conditions, e.g. entry speed to one of the individually assessed bends, would also be unreliable and should be measured directly instead of being simulated based on the track model's geometry.

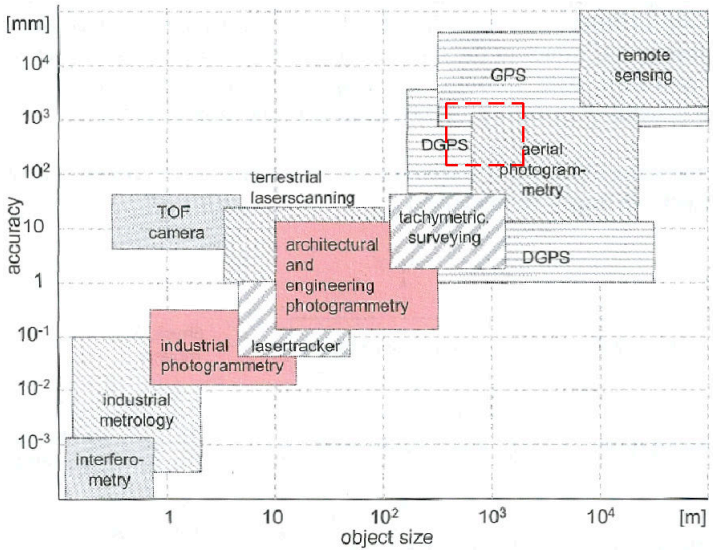


Figure 3.3: Relationship between object size, accuracy requirements, and measurement method; adapted from Luhmann et al. (2014, p. 4, fig. 1.4).
- - - Track size and reconstruction accuracy requirements.

The achievable accuracy with respect to the size of a scene or an object, using existing measurement methods, is illustrated in figure 3.3. The apparent positive linear correlation between object size and accuracy is indicative of the fact that the choices of the appropriate method for a task are limited for certain combinations of these two factors. In theory, objects the size of a track are not within the domain of applicability of terrestrial close-range photogrammetry. Nevertheless, the alternative methods of aerial photogrammetry, differential GNSS, and tachymetric surveying, which can yield accuracy in the range of 1 cm to 1 m, are not an option in this setting mainly for two reasons: firstly, because of the fact that the tracks are for the most part covered by roofs and blinds; secondly, due to the requirement for a workflow that can be fully integrated into the standard practice of a track walk. The former renders them GNSS-denied environments and not particularly suited for unmanned aerial vehicles even if unrestricted access was granted to a survey crew and athletes and maintenance workers were prevented from entering the site. The latter excludes traditional surveying methods that require expensive and bulky equipment, such as terrestrial laser scanners, as they would require at least a couple of days on site (SAIT, 2012) and would also struggle with the narrow space and glossy surfaces. The limited time-frame, in combination with the restrictions imposed on what is allowed during a track walk, means that

site availability

scene signalisation

setting out artificial targets or tampering with the ice surface to facilitate direct georeferencing or image matching is not an option either. The absence of prior knowledge in object space means that for the image orientation and sparse reconstruction process it suffices to describe the scene with a discrete set of 3D points without further assumptions about them lying on a plane or a spatial curve (Förstner et al., 2016, p. 442). The transformation of each one of the acquired images to a symbolic representation comprising 2D points—called feature points—and the problem of finding the correspondence between features of the same object point projected on different images are discussed in section 3.3.2.

scene model

image model

3.2.2 Measurement Mode

The photogrammetric processing of the acquired images can be performed either online or offline (Luhmann et al., 2016, pp. 134-136). Offline photogrammetry allows the use of non-metric cameras, as the reconstruction and camera interior and exterior orientation can be estimated at the same time in order to acquire more accurate estimations; only the acquisition time constrains the amount of data that will be collected. In contrast, online photogrammetry requires high-end metric cameras with stable geometry that tend to be more expensive. In the computer vision community, online systems based on monocular or multi-stereo vision provide the level of abstraction and automation that is desirable but, as they are mostly developed for robotics applications with real-time performance as a priority—i.e. simultaneous localisation and mapping (SLAM) (Dissanayake et al., 2001; Pollefeys et al., 2004; Durrant-Whyte et al., 2006; Davison et al., 2007; Pollefeys et al., 2008; Newcombe, 2012; Fuentes-Pacheco et al., 2015) or visual odometry (VO) (Nister et al., 2004; Konolige et al., 2011; Scaramuzza et al., 2011)—and not dense surface reconstruction, they do so in the expense of image resolution, object-space point density, and, consequently, reconstruction accuracy. Such methods are not considered herein but a commercially available hardware solution, based on their principles, for mapping applications is described in section 3.6.2 and evaluated in section 10.3.

A compromise between the two measurement modes can be achieved by providing online feedback to the user that controls the sensor, in order to make sure that the captured images are sufficient for offline reconstruction. Such a human-in-the-loop process was described in Hoppe et al. (2012) and Rumpler et al. (2017). In contrast to most VO and SLAM algorithms, which rely on high frame rates and minimal camera motion between successive images, they used wide-baseline still-images with 10 MP resolution. Their assumption was that the images were sequentially captured and had sufficient overlap so that each image could be processed online. The validation

online feedback

of each new image required up to 2 seconds on a 3.2 GHz laptop processor and was based on image overlap, ground resolution, and ground coverage (Rumpler et al., 2017, p. 260). Mostegel et al. (2016) proposed a pre-trained prediction module based on machine learning techniques, in order to assess whether or not a newly captured image by a unmanned aerial vehicle (UAV) was good enough for multi-view reconstruction. The confidence prediction for 16 MP images was performed on a laptop and required approximately 2 seconds per image.

image quality

In videogrammetric surveying applications, key-frame selection is usually the first step in the post-processing stage and is based on one or a combination of metrics such as sufficient overlap, sufficient baseline, minimal reprojection errors, ratio of the number of correspondences to the number of features, relative orientation error, or the distribution of correspondences over the frames (Seo et al., 2008; Rashidi et al., 2013). Torr et al. (1998) proposed a general metric calculated as the weighted sum of the number of inliers and outliers, the errors and residuals aiming at avoiding degenerate imaging configurations but the user-defined weights and thresholds for each one of the aforementioned factors was project-specific.

The quality of the image, with respect to motion or out-of-focus blur, has also been used in order to remove blurry frames that may affect the quality of the reconstruction. Motion blur due to camera movement can be a problem when a hand-held camera is used in poorly lit environments. In general, achieving a sharp and well-exposed image depends on the selection of several camera parameters which include the ISO speed, exposure time, and aperture size; the image quality relies on the photographer's skill and steady hands. However, for a 3D reconstruction task, the requirement for a stable camera geometry and a deep field of view in combination with the lens's specifications call for a fixed-focus lens and an aperture-priority mode for image acquisition, in which case a good balance needs to be found between a high ISO number, which may induce noise, and a high exposure time, which may induce motion blur. A rule of thumb for hand held photography on a full frame sensor without image stabilisation and a lens with a focal lens of F mm is achieving an exposure time of at most 1/F sec.

Identifying sharp images is easy for humans but doing so automatically without a reference image is quite hard and is an active research field with respect to UAV imagery (Sieberth et al., 2015). In a review by Pertuz et al. (2013), 36 focus-measure operators were assessed for 3D reconstruction based on shape-from-focus. It was concluded that the operators based on the Laplacian of the image, e.g. the variance of an approximation of its second derivative (Pech-Pacheco et al., 2000), have the best performance but depends on the imaging conditions and is sensitive to image noise. Rashidi et al. (2013) used a no-reference image quality metric proposed by Crété-Roffet

et al. (2007) which assigned a value between 0 and 1 to each image, based on how much its high frequency content was affected after the application of a smoothing filter. Their method was based on the assumption that sharp images would be more severely affected compared to images that were already blurry. The effectiveness of the method was assessed with respect to the correlation of the results to human assessments.

Charrier et al. (2015) compared seven no-reference image quality metrics, including the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) (Mittal et al., 2012) and the Natural Image Quality Evaluator (NIQE) (Mittal et al., 2013), which compare an image to a model computed from images of natural scenes in order to estimate their quality. Their assessment, which was focused on lossy image compression techniques, concluded that not all metrics are equally sensitive to the same quality degradation and that their performance depends on the image content.

additional measurements

Even if a successful alignment of the acquired images can be confirmed on site and yield a complete and precise reconstruction of the track's shape and form, the scale and orientation of the model cannot be defined based solely on photogrammetric measurements, q.v. section 3.3. Inaccurate estimation of its scale or orientation may render the track model unfit for the purpose of simulating the movement of an object on top of it under the influence of gravity. Thus, additional information for the datum definition are required; it can come either in the form of ground control points (GCP), i.e points with known object space 3D coordinates that can be measured on the images, or from direct measurements of the exterior orientation during image acquisition. Given that GNSS may only be used on the parts of the track that are not covered, for most tracks this means the start and finish areas. This may leave nonlinear model distortions unaccounted for but could be a viable option for the datum definition, considering that as few as three GCPs are necessary. Using aerial or satellite images and digital elevation models to collect GCPs can be used to a certain extent but identifying the selected points on the terrestrial images could be challenging due to the differences in image resolution and perspective.

Alternatively, combining a hand-held camera with an inertial measurement unit (IMU) could provide the necessary information, as the two sensor units are complementary and can potentially compensate for each others deficiencies, but can effectively do so only when fast moving sensors are considered for a short period of time (Strelow et al., 2004; Corke et al., 2007). An IMU typically combines three orthogonal accelerometers that measure linear acceleration and three orthogonal gyroscopes that measure angular velocity. The position and orientation of the unit can be obtained by integration but the noise level in the measurements of low-cost portable IMUs suitable

for hand-held mobile systems tend to accumulate over time and lead to significant errors, even if the system is thoroughly calibrated. In order to avoid the complexities of a system calibration, Jung et al. (2001) independently estimated the trajectories of the camera and the IMU offline; they assumed that the camera motion was smooth along small trajectory segments and fitted a second order polynomial to the sensor coordinates over time. This approach was extended for online use, in order to estimate the scale of a monocular SfM solution, by introducing a small time lag. However, scale estimation errors up to 50% were reported (Nützi et al., 2011, p. 292). The solution drift can be mitigated if sensors such as magnetometers or a barometer are added to the IMU, but this can make the tedious visual-inertial system calibration and spatio-temporal alignment (Lobo et al., 2007; Mair et al., 2011; Hesch et al., 2013) even more complicated.

3.2.3 Surface Matters

ice

The surface of a solid object can be defined as the simplest type of interface that separates it from its surroundings, e.g. air or vacuum (Luth, 2001). Ice is the solid state of water but the very definition of its surface is not simple. Its quasi-liquid nature, even though it was observed almost two centuries ago (Faraday, 1859), remains a puzzle (Hobbs, 2010, p. 392; Petrenko et al., 1999, p. 227; Rosenberg, 2005; Kietzig et al., 2010). There are 17 known crystalline phases of water (Falenty et al., 2014). The normal and most common type of ice—Ice I_h—is created in two ways. When water condenses from vapour at temperatures between 0–100°C, it creates single crystals in a variety of shapes forming frost or snow. If snow survives long enough, it turns into firn and then to polar or glacier ice under the pressure from the accumulated layers. The second way, is by freezing liquid water under normal atmospheric pressure (Fletcher, 1970; Petrenko et al., 1999; Hobbs, 2010), which is how ice is applied on sliding tracks (Poirier et al., 2011c).

shape

level of detail

The shape of an object is not a mathematical idea; its perception, rather than definition, depends not only on our expectations—in other words the scene model, q.v. figure 3.1—but also on its physical properties and the method which has been chosen to interact with it (Koenderink, 1990). The ambiguity stems from the fact that surface definition in general is scale-dependent and in some cases not clear, e.g. snow, sand, clouds (Förstner et al., 2016). If a surface is treated as a continuum of spatial wavelengths, the longer ones describe the surface form—the underlying shape—and the shorter ones its texture (Leach, 2013). The former, large-scale characteristic geometric layout of the surface, is of interest with respect to the attitude of objects the size of a sled, i.e. its position and orientation on the track. Its accuracy affects the estimation of the normal force acting on the sled due to gravity. The latter, small-scale characteristic texture, defines whether it

is smooth, or rough, or anything in-between and how it interacts with light (Gibson, 1979).

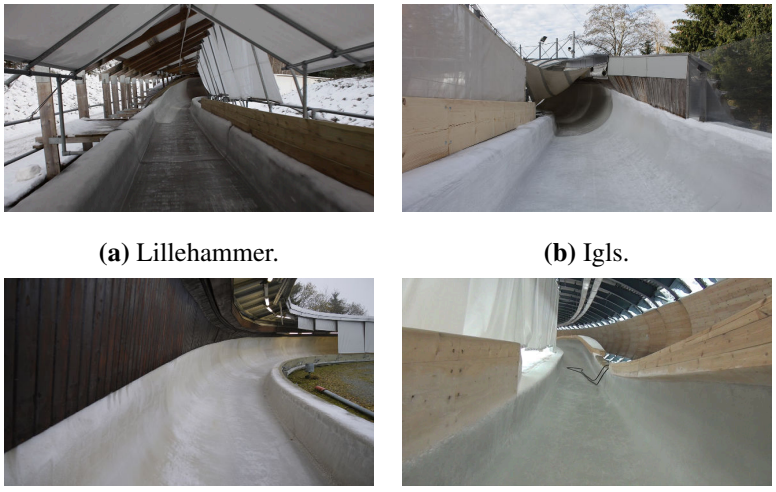
roughness

Surface roughness is of interest with respect to the friction between the steel sled runners and the ice. An important parameter of the sled's motion model, q.v. section 2.2.2, is the contact area between the runners and the ice surface (Poirier et al., 2011b). Nevertheless, as there is no generally accepted theory on the mechanism of friction (Kessler, 2001; Popov, 2016), most engineering applications adopt empirical simplifications stating that the force of friction is proportional to the normal force and independent of the area of contact. The friction coefficient that expresses this proportionality is experimentally estimated as the ratio of the shear strength of the surface asperities to the material's hardness (Hobbs, 2010, p. 413).

On the one hand, the grooves or knives at the bottom of the runners, which invariably cut into the ice during steering, have to be smaller than 2 mm according to IBSF (2016, p. 32) regulations. On the other hand, the ice surface is not smooth either because of the way it was prepared, the weather conditions during that time or afterwards, or because of it being manually scrapped to bring it to the desired shape. In any case, the size of the asperities that define the contact area is at submilimetre level (Bromley, 1999). In order to describe the ice surface at this scale, its texture can either be sampled locally by appropriate optical (Leach, 2011) or tactile surface metrology methods or considered to be a statistical parameter and simulated as such (Koenderink, 1990, p. 33; Nayar et al., 1991, p. 612). It is obvious that reconstructing an object of the size of a track at that level of detail is impractical. In practice, experimental tribology studies (Poirier et al., 2011a; Tomaszewska et al., 2017) are conducted to estimate a friction coefficient that can adequately describe the specific operational condition—e.g. time, weather and speed—encountered by a specific type of sled at each track as it not only depends on materials and surfaces (Bhushan, 2013, p. 264).

reflectance

Apart from the ice layer, the materials comprising the track's immediate environment are concrete, wood, plastic or fabric, and steel—q.v. figure 3.4. They are all widely used in infrastructure works and in general, unless they are specially treated, their surface texture is favourable to optical measurement techniques. The discussion gets a lot more complicated when translucent or transparent objects are considered as the assumption that the measurements made by optical methods are actually made on the desired surfaces becomes uncertain. The question is whether the measuring process and the sled interact with the same surface or not. There is no doubt about the tip of the probing stylus of a profiler; however, this is not the case when it comes to non-contact, in particular optical, measurement methods (Blais, 2004) that rely on the Lambertian reflectance model (Nayar et al., 1991).



(a) Lillehammer. **(b)** Igls.

(c) Winterberg. **(d)** Sochi.

Figure 3.4: Materials used in sliding track construction.

As ice is not a common surface material in close-range photogrammetry, only an indication of the accuracy potential of range measurements on similar surfaces—i.e. snow or marble—can be acquired from engineering applications. For instance, studies on snow thickness estimation using terrestrial laser scanners report 10 cm errors (Prokop, 2008, p. 162; Deems et al., 2013, p. 475; Jaakkola et al., 2014, p. 590) whereas the bias on marble surfaces can be up to 4 mm (Godin et al., 2001; Guidi et al., 2009). The characterisation of the reflectance properties of ice or snow surfaces is quite complicated and cannot be easily standardised mainly because of the fact that its condition may vary significantly with respect to grain size, water and impurities content, temperature etc. For instance, frosty ice exhibits diffuse characteristics—just like fresh snow—and the amount of air bubbles or debris also affect light absorption and forward or backscattering (Hobbs, 2010, p. 207; Petrenko et al., 1999, p. 216; Pope et al., 2014, p. 4). An in-depth investigation of reflectance models and the optical properties of ice are beyond the scope of this work—for more details on the subject see Warren (1982), Nayar et al. (1991) and references therein. Methods for the 3D scanning of surfaces with more complex optical characteristics or through media which do not transmit light rays in straight lines are still in their infancy and require considerable effort (Ihrke et al., 2010) with unpredictable results.

3.3 Estimation Theory and Practice

In surveying in general, and particularly in photogrammetry, the quantities of interest are very rarely measured directly. For example, instead of a distance between two points in space, the time of

flight of an electromagnetic pulse or two angles from a baseline with known length are measured. As the scene model in this setting was defined by a set of 3D coordinates, q.v. section 3.2.1, these are the parameters that need to be estimated together with the parameters of the cameras' exterior orientation—and sometimes the interior orientation, as well—from observations of their projections on images. In most photogrammetric applications, the number of observations can be several orders of magnitude more than the unknown parameters. They also contain errors, so they are treated as uncertain quantities with stochastic properties in adjustment methods. This redundancy leads to over-constrained problems with multiple solutions.

Estimation theory provides the tools for the determination of an optimal unique solution from a set of redundant and inconsistent measurements. The mathematical model that is used to tackle such problems is an abstraction that attempts to describe the physical processes involved in measurements. Its complexity is a design choice that reflects our understanding of the process or compromises for economy and efficiency. It comprises a functional and a stochastic model and aims to find estimates for the true values of the parameters and a compatible set of observations from the original set of uncertain observations (Mikhail et al., 1976, pp. 5–8; Förstner et al., 2016, pp. 75–76). The functional model describes the deterministic relationship between the observations and the parameters. This estimation problem can be formulated based on the Gauss-Markov model that describes each observation as a function of the parameters and yields a set of indirect observation equations (Mikhail et al., 1976; Cross, 1983; Luhmann et al., 2014; Förstner et al., 2016). This distinction between observations and parameters is simple, reasonable, and intuitive but cannot handle constraints between observations and parameters where essentially all the variables are treated as observations. Nevertheless, the Gauss-Markov model is flexible enough to handle most photogrammetric problems (Mikhail et al., 2001, p. 396; Kraus, 2007, p. 441; McGlone, 2004, p. 81) and can be extended to incorporate constraints between the parameters that can be treated as additional uncertain observations. For the unified approach based on the Gauss-Helmert model of condition equations that treats all the parameters as, potentially stochastic, observations see Mikhail et al. (1976, p. 333) and Förstner et al. (2016, p. 81). The stochastic properties of the observations are in general unknown but their complete probability distribution is not required; their uncertainty is conveyed by the covariance matrix which needs only be defined up to a scale (Mikhail et al., 1976, p. 65; Förstner et al., 2016, pp. 89–90).

mathematical model

functional model

stochastic model

The scene and image models that are used in photogrammetry simplify the mathematical formulation of the problem but have some implications with respect to the datum definition as the scene-points' coordinates cannot be defined by their image projections alone.Seven datum para-

datum definition

meters are required in order to define the object space coordinate system; one for the scale, three for orientation, and three for the origin. This can be achieved by a minimal parameterisation, e.g. by replacing seven appropriately chosen point coordinates with their known values. Alternatively, an additional set equations of the form can be introduced. This constraint equation-set could include, for example, three equations for the position of a point and another two for the measured vertical distance and horizontal directions between two points, respectively. If the absolute georeference of the reconstruction is not of interest, then the origin and horizontal orientation of the reference system can be arbitrarily set and only a single vertical distance, e.g. between two cameras, is required for the datum definition.

inner constraints

Each combination of available information for the definition of the datum parameters yields a different estimation of the unknown parameter corrections and their covariance matrix. Nonetheless, a free network adjustment can be performed with a set of seven special— in the sense that it minimises the trace of the aforementioned covariance matrix—constraint equations called inner constraints (Granshaw, 1980; Fraser, 1982; Dermanis, 1994) which define a best-fit similarity transformation between an arbitrary photogrammetric coordinate frame and the one defined by the approximate object space coordinates of the points—and possibly the camera projection centres.

3.3.1 Self-Calibrating Bundle Adjustment

analytical model

The self-calibrating bundle adjustment was introduced more than 50 years ago (Brown, 1958; Schmid, 1958; Brown, 1971) and is the most widely used analytical model in photogrammetric computer vision (Triggs et al., 1999, p. 299) for the optimal image-based reconstruction of a 3D scene due to its flexibility and increased accuracy (Granshaw, 1980, p. 181; Cross, 1983, p. 106; Förstner et al., 2016, p. 643), even for real-time applications (Engels et al., 2006). It is used in order to simultaneously estimate the cameras' interior and exterior orientation parameters and the scene point coordinates from the observations of their image projection coordinates.

collinearity equations

The functional model of the bundle adjustment method is based on the collinearity equations 3.1 and 3.2. They express the transformation of an object point P with coordinates $[X_P, Y_P, Z_P]^T$ to its observed projection on an image $[x'_{p'}, y'_{p'}]^T$, see figure 3.5, as a function of the corresponding camera's interior and exterior orientation parameters. Thus, each point observation on an image provides two observation equations:

$$x'_{p'} = x_{pp} - c\frac{r_{11}(X_P - X_{PC}) + r_{21}(Y_P - Y_{PC}) + r_{31}(Z_P - Z_{PC})}{r_{13}(X_P - X_{PC}) + r_{23}(Y_P - Y_{PC}) + r_{33}(Z_P - Z_{PC})} + \Delta x_p \tag{3.1}$$

$$y'_{p'} = y_{pp} - c\frac{r_{12}(X_P - X_{PC}) + r_{22}(Y_P - Y_{PC}) + r_{32}(Z_P - Z_{PC})}{r_{13}(X_P - X_{PC}) + r_{23}(Y_P - Y_{PC}) + r_{33}(Z_P - Z_{PC})} + \Delta y_p \tag{3.2}$$

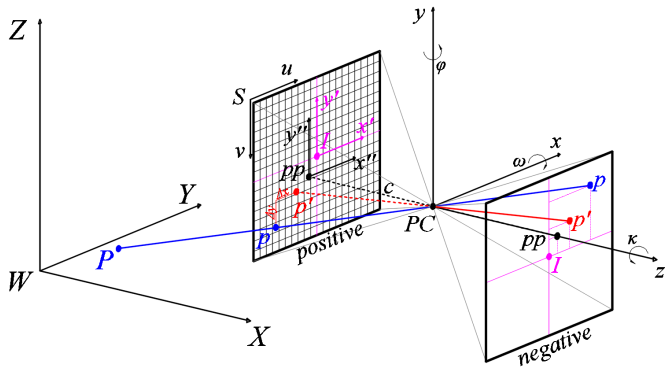


Figure 3.5: Photogrammetric coordinate systems; adapted from Luhmann et al. (2014, pp. 144,148,258; fig. 3.41,3.49 and 4.4). The rotation angles $\omega - \phi - \kappa$ are defined counter-clockwise looking towards the origin from the positive axes, following the right-hand rule.

where r_{ij} are the elements of the camera rotation matrix $\boldsymbol{R}_{\phi\omega\kappa}$, q.v. equation 3.3, which can be defined as the rotational transformation from the camera to the object coordinate system using a sequence of three Euler angles around the camera's moving axes, and Δx_p, Δy_p are the corrections according to the standard lens distortion model (Brown, 1971) which considers 3 polynomial coefficients for radial (K_1, K_2, K_3) and 2 for tangential (P_1, P_2) lens distortion, q.v. figure 3.5.

$$\boldsymbol{R}_{\phi\omega\kappa} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\omega) & -sin(\omega) \\ 0 & sin(\omega) & cos(\omega) \end{bmatrix} \begin{bmatrix} cos(\kappa) & -sin(\kappa) & 0 \\ sin(\kappa) & cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} cos(\phi)cos(\kappa) + sin(\phi)sin(\kappa)sin(\omega) & cos(\kappa)sin(\phi)sin(\omega) - cos(\phi)sin(\kappa) & sin(\phi)cos(\omega) \\ cos(w)sin(\kappa) & cos(\kappa)cos(\omega) & -sin(\omega) \\ cos(\phi)sin(\kappa)sin(\omega) - cos(k)sin(\phi) & sin(\phi)sin(\kappa) + cos(\phi)cos(\kappa)sin(\omega) & cos(\phi)cos(\omega) \end{bmatrix} \quad (3.3)$$

covariance

The effect of the accurate description of the stochastic model on the quality of the adjustment results has long been established in photogrammetric applications. Nevertheless, it is convenient to assume that they are uncorrelated and even of equal variance because it is hard to estimate the correlation and it simplifies the calculations (Mikhail et al., 1976, p. 119; Granshaw, 1980, p. 188; Cross, 1983, p. 39; Förstner et al., 2016, p. 85). In practice, the statistical properties of image point measurements that are performed manually or automatically at artificial targets or at image intensity structures with a distinct centroid, e.g. circles or ellipses, are estimated from previous experience or from data redundancy (Mikhail et al., 1976, p. 7; Cross, 1983, p. 33). It has been demonstrated that their accuracy, for targets with size of 5-15 pixels, can be better than 0.1 pixels (Luhmann et al., 2014, p. 463). The subject of feature localisation accuracy with respect to non-

signalised scenes is discussed in section 3.3.2 in the context of feature detection and matching. Even though it has been observed that the uncertainty of points tracked over video frames can be correlated (Sugaya et al., 2003, p. 606; Kanatani, 2004, p. 1309), for still frames it can be assumed that there is no correlation between different points.

quality

The free network solution can also be used to verify the internal quality of the model, as it is not influenced by inconsistencies between the control points used and is only affected by the photogrammetric observations. However, the object points' standard deviations that were estimated from a free-network solution should not be used for the accuracy assessment of a photogrammetric result, which can only be reliably assessed against reference points that have been measured independently with higher accuracy and have not been used in the bundle adjustment. As the errors in control points can be interpreted as observation errors, which makes them difficult to detect (Atkinson, 1996, p. 39), their accuracy should be 5–10 times better than what can be achieved by the photogrammetric measurements (Luhmann et al., 2014, p. 341,347,359,553). The precision of the bundle adjustment results can be evaluated in a number of ways. A global check of the reference variance, which should be close to 1, gives an indication of how well the data fit the mathematical model. The a posteriori (co)variances and the associated error ellipses can provide useful information on the estimation uncertainty of each parameter. The residuals of image and object points, which should be small and homogeneous, can be used in statistical tests, in order to detect outliers under the rather restrictive assumption that only a single outlier exists in the observations with a relatively small error (Baarda, 1973; Cross, 1983, p. 40; Cooper et al., 1988; Luhmann et al., 2014, p. 101; Förstner et al., 2016, p. 689).

practical problems

The practical problems of self-calibration are related to the fact that the interior orientation parameters are correlated with the exterior orientation and between them. If a sufficient number of convergent images is not available, then the principal distance can only be reliably estimated if the points are well distributed in 3D space; in order to reliably retrieve the principal point coordinates variable roll angles could be used, instead. Reliable object space coordinates can be estimated within a self-calibrating bundle adjustment even if these correlations are ignored, however, the interior orientation should not be used as fixed parameters in a subsequent bundle adjustment (Gruen et al., 2001, pp. 110–112;Luhmann et al., 2014, pp. 162,339,576–578).

critical configurations

Care should be taken in order to avoid imaging configurations that are known to yield poor results when the bundle adjustment is used for euclidean reconstruction and/or self-calibration. Those that require both the scene points and the camera positions to lie on a critical surface, e.g. a cylinder, are easily avoidable in practice (Horn, 1988; Kahl et al., 2002; Luhmann et al., 2014,

p. 306). Certain camera motions such as pure rotation, translation, or orbital motion should also be avoided (Sturm, 1997; Sturm, 2000; Sturm, 2002; Kahl, 1999; Kahl et al., 2000; Wu, 2014; Förstner et al., 2016, p. 453); to a certain extend their effect can be mitigated by fixing some of the interior orientation parameters i.e. skew and aspect ratio (Triggs, 1997).

3.3.2 Feature Detection, Description, and Matching

The bundle adjustment is a unified framework that combines certain well-understood processes in surveying, i.e. triangulation, resection, calibration. The main challenges pertain mostly to practical issues with respect to automating the entire process, as a few millions unknown parameters need to be estimated from at least twice as many observations (Luhmann et al., 2014, p. 323). In order to analyse the acquired images using the bundle adjustment method, according to the metamodel in figure 3.1, an appropriate image model is necessary; one that can reduce the image intensity samples taken at millions of points on the scene to a symbolic representation comprising a set of 2D pixels on each image (Förstner et al., 2016, p. 448).

image model

Automatically identifying the 2D projections of specific 3D locations on objects in the scene which are of interest to the application at hand—line features are not that useful during the image orientation phase (Hartmann et al., 2016, p. 50)—would be ideal; nevertheless, doing so on images taken from various viewpoints under different illumination conditions, without using artificial high-contrast targets, is an almost impossible task (Kanatani, 2004, p. 1308). In close-range photogrammetry, algorithms that attempt to detect stable and distinctive 2D feature—or interest or tie—points across the image set are used for this purpose. The feature points need to be locally described in a unique way, based on properties related to the image pattern at their immediate neighbourhood (Tuytelaars et al., 2007, p. 178), so that they can be automatically matched in order to establish the point correspondences that will create the system of collinearity equations.

A feature point needs to have certain characteristics in order to be considered good (Tuytelaars et al., 2007; Wolfgang et al., 2009). Repeatability across many images is essential and is achieved through robustness or invariance to deformations of the image patterns. A feature point has to be distinctive but this is often a factor that competes against locality—consider, for example, repetitive patterns. Interpretability is a desirable attribute—in the sense that features detected on patterns that have a canonical geometric appearance, i.e. corners or circles, can be visually verified by a human operator—but not essential. The total number of feature points should also be considered, as enough correspondences need to be established, but computational efficiency is also critical. The increasing number of images commonly used in an image-based modelling project

repeatability

interpretability

quantity

may render the matching process the bottleneck of the entire reconstruction process but there are several ways to overcome this problem. For photogrammetric applications, image acquisition is orderly planned or the approximate camera exterior orientation is directly measured so that an indication of overlapping views between images is available. For generic unordered image collections, the search space for candidate images where a point match may exist can be limited by reducing the number of the detected feature points, the number of images, or the total number of potential matches (Hartmann et al., 2016, p. 48). More importantly, feature points should be accurately localised. The term localisation accuracy is used to express the requirement for a feature point detected in several images to correspond to the same object location, whichever location that is; unlike the majority of the literature, this is described as the tracking error in Morris et al. (2002, p. 108).

localisation accuracy

Many studies have been published on comparisons between a plethora of feature detector and descriptor combinations. Schmid et al. (1998) compared several detectors, including the popular Harris et al. (1988) corner detector and the Förstner et al. (1987) operator, based on repeatability under various image transformations. Mikolajczyk et al. (2004) reviewed scale and affine invariant detectors such as the Harris-Laplace. Fraundorfer et al. (2004) compared detectors on images of non-planar scenes and found that the performance of MSER (Matas et al., 2002) was the best; however, its superiority could only be confirmed for flat surfaces (Fraundorfer et al., 2005; Moreels et al., 2007; Dickscheid et al., 2009; Aanes et al., 2012). Moreels et al. (2007) reported the good performance of the SIFT descriptor (Lowe, 2004) based on results from image sequences of 3D objects under varying viewing angles and illumination. According to Juan et al. (2009), SURF (Bay et al., 2008) is faster than SIFT with equal performance for small viewpoint and illumination changes. Aanes et al. (2012) concluded that the Harris corner detector, the Hessian blobs (Lindeberg, 1998; Mikolajczyk et al., 2005) and the Difference of Gaussian (DoG) (Lowe, 2004) where the best among ten detectors including the FAST corner detector (Trajković et al., 1998) which was unreliable based on the recall rate—the ratio between potential matches and the total number of features. The performance of the Harris, DoG, and Hessian detectors and the comparable effectiveness of the SIFT and SURF descriptors was also confirmed in Yong et al. (2015). In order to assess the effectiveness of features with low computational requirements Heinly et al. (2012) compared the binary descriptors BRIEF (Calonder et al., 2010), ORB (Rublee et al., 2011), and BRISK (Leutenegger et al., 2011). Their usefulness in time-critical applications was assessed in Miksik et al. (2012) who also claimed that descriptors such as LIOP (Wang et al., 2011), MRRID and MROGH (Fan et al., 2012) can outperform SIFT and SURF in precision and recall.

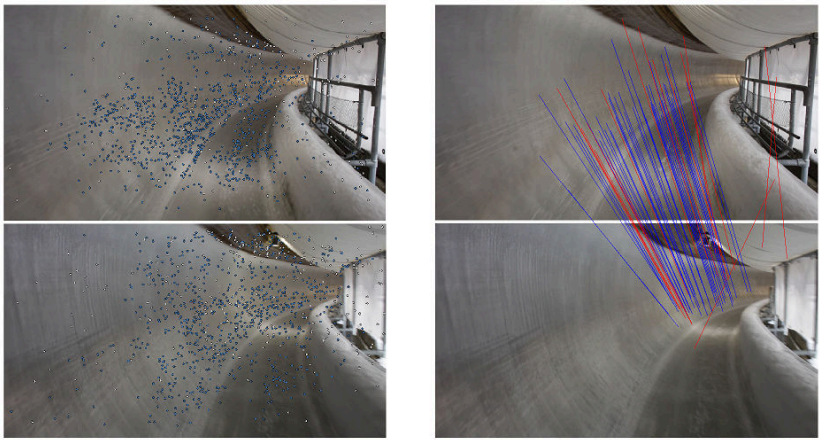


Figure 3.6: Feature point detection (left column) and matching (right column) using PhotoScan. The blue points and lines indicate successful matches whereas red lines indicate feature points in the top row image erroneously matched to points in the bottom image.

optimality

New detectors and descriptors are proposed almost annually accompanied by comparative evaluations to previously proposed ones on various applications (Apollonio et al., 2014; Schmidt et al., 2015). So do benchmarks and evaluation criteria that question the validity of the results in situations other than those used within each experimental set up. For example, Cordes et al. (2013, 2015) identified performance differences when high-resolution images are used in the evaluation and Rey-otero et al. (2015) suggested that the repeatability criterion favours certain types of detectors. As the verdict on the most effective and efficient detector-descriptor combination for accurately localised feature points for camera calibration and image-based reconstruction is still out, the best strategy for non time-critical applications is to use a variety of features with complementary characteristics (Förstner et al., 2004; Tuytelaars et al., 2007; Dickscheid et al., 2009).

approximate values

The most challenging part of the bundle adjustment process, which is also very time consuming, is generating approximate values for all the parameters in order to initialise the iterative solution. The cameras' exterior orientation and the object space coordinates of the scene points that were observed on the images need to be estimated from the features that were matched based on their descriptors and may potentially include many false matches. In the case of a stereo-pair, an aerial survey, or any canonical normal configuration, the generation of approximate values for the exterior orientation, which can facilitate estimating the image points correspondences, is relatively straightforward. However, this in not the case for most close-range photogrammetric networks, unless the cameras are set-up over known imaging stations or an adequate number of coded targets with known object space coordinates is used. In general, the required robustness in

finding correspondences between image points is achieved by extending the search domain from feature space, which is based on the descriptor vector, to image and object space where additional constraints can be imposed. For automated target-free workflows, the approximate camera orientation parameters and the object space coordinates of feature points, which are observed on the images and serve as tie points, are estimated by resections followed by spatial intersections or by relative orientation of image pairs or triplets.

Several sequential (Förstner et al., 2016, pp. 708–709) or hierarchical (Fitzgibbon et al., 1998) SfM strategies, attempting to detect and eliminate gross errors in the observations resulting from false matches, have been proposed but no general technique is available (Mikhail et al., 1976; Kraus, 1997, p. 43; Luhmann et al., 2014, p. 323; Förstner et al., 2016, p. 440). In a recent review on large-scale tie-point matching, Hartmann et al. (2016) suggested that as memory requirements will keep increasing, new methods will need to emerge; for instance, Novak (2014, p. 14) reported that SIFT implementations can only handle images with resolution up to 6 MP. The main problem they identified was that wide-baselines with viewpoint changes greater than 30° pose a significant challenge even for scale- and affine-invariant descriptors (Tuytelaars et al., 2007, p. 258; Moreels et al., 2007, p. 28; Hartmann et al., 2016, p. 61).

3.3.3 Dense Image Matching

The landscape for dense image matching, which aims to estimate depth information at every pixel in each image, is even more complicated than what has been described in section 3.3.2 with respect to the abundance of algorithms that attempt to solve the correspondence problem. Even though several advancements have been observed, this is still an open problem; for an account of the developments of image matching in photogrammetry see Szeliski (2011), Gruen (2012), and Remondino et al. (2014). Dense image matching methods can be categorised according to the number of images taken into account; the process is referred to as stereomatching, if two images are considered, or multi-view stereo for more than two images. According to the extent of the area around the pixel of interest that is used in order to establish the correspondences, local methods use small areas around each pixel. They are computationally efficient but are affected by occlusions and regions with poor texture. On the other hand, global methods take into account entire lines or the full image and can be more robust but slower (Brown et al., 2003).

The collinearity equations 3.1 and 3.2 express in mathematical terms that, in theory, a camera's perspective centre, an object-space point's projection on the image, and the point itself are collinear. Another condition, useful in image matching, is the coplanarity condition which expresses

the fact that an object-space point p projected on two images lies on the same plane as the camera perspective centres C' and C'' and its image projections p' and p'', with image coordinates $\boldsymbol{x}'$ and $\boldsymbol{x}''$, respectively. If the cameras' interior and exterior orientation parameters are known, then the fundamental matrix $\boldsymbol{F}$ which encapsulates their relative orientation can be calculated. Searching in the second image for the conjugate p'' of a point p' that was identified in the first image, in order to triangulate its object-space position, can be limited close to a line called the epipolar line, q.v. figure 3.7. The coplanarity condition is expressed as $[\boldsymbol{x}'1]^T \boldsymbol{F}[\boldsymbol{x}''1] = 0$ (Luhmann et al., 2014, p. 310). The remaining problem is defining an appropriate disparity range in order to reduce the search length along the epipolar line.

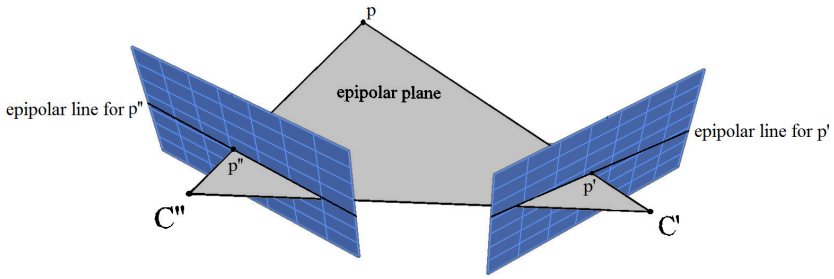


Figure 3.7: Epipolar geometry.

For the estimation of the matching score between the image patch around the point of interest and the patch around a candidate location for its projection on a second image, metrics such as the sum of absolute differences between the pixel values or normalised cross correlation are used. Similarity transformations between the patches need to be taken into account explicitly (Lewis, 1995). Alternatively, least-squares template matching (Gruen, 1996) can be used in order to model radiometric and affine distortions (Luhmann et al., 2014, pp. 452–454).

Several algorithms are being tested every year on the stereo and multi-view benchmarks introduced by Scharstein et al. (2002) and Seitz et al. (2006), respectively. In addition, results from their performance on specific photogrammetric applications are continuously reported. For example, Jaud et al. (2016) concluded that the commercial PhotoScan (2018) software appears to perform better than the open-source MicMac (Pierrot-Deseilligny et al., 2006) on rough terrain modelling from UAV images, but this is partly accounted to the image orientation procedure and camera calibration model. In Remondino et al. (2014), four algorithms that are considered state-of-the-art in dense image matching—namely SURE (Rothermel et al., 2012), MicMac, PhotoScan, and PMVS (Furukawa et al., 2010)—were evaluated on five close-range datasets. Their main observations was that, apart from problems related to image acquisition and content, the results are sensitive to the user-defined parameters, many of which do not have a clear interpretation and

their effect on the results can be unpredictable. They refrained from ranking the four algorithms as there is still room for further developments in the field, whereas Hosseininaveh Ahmadabadian et al. (2013) favoured PMVS, which was compared to MicMac and several implementations of the semi-global matching algorithm (Hirschmuller, 2005, 2008) for the reconstruction of small objects using a stereo camera rig.

3.4 Photogrammetric Network Design

Given the initial plan for a photogrammetric project, q.v. section 3.2, the most challenging part of the process is the design of an observational scheme, which comprises the imaging network and object points, that can meet the accuracy requirements, especially in the context of a self-calibrating bundle adjustment, q.v. section 3.3.1. The total number of images and the cameras' exterior orientation at the time of acquisition, together with the size and distribution of the target array directly affect the network's quality, as they define the total number of observations and the angles of convergence and incidence, which in turn affect the triangulation accuracy. The interactions between these factors are hard to control, as they depend on the geometry of the object and its surrounding environment which—in combination with several parameters of the imaging system such as the sensor size, image resolution, lens focal length and aperture size—affect the depth of field, image scale, and field of view. A compromise is required between all the aforementioned factors in order to achieve an acceptable level of image measurement accuracy. (Marshall, 1989, p. 89; Mason, 1995, pp. 52–54; Fraser, 1996, pp. 263–264; Mason, 1997, p. 1095; Luhmann, 2011; Luhmann et al., 2014, pp. 545–547)

design constraints

It is quite clear that cause and effect relationships cannot be easily established; coming up with a single empirical formula that expresses the attainable quality as a function of the aforementioned parameters, in order to solve the design problem in one step, has been problematic (Parian, 2007). Therefore, the most common approach for the photogrammetric network design process is based on its decomposition in four stages (Fraser, 1984, 1996), initially proposed for geodetic networks by Grafarend (1974):

design stages

- Zero-Order Design (ZOD): the datum problem

  fixing the datum defect by choosing a set of minimal parameter constraints will either affect the a posteriori precision estimations of the object point coordinates, relatively to their position with respect to the constrained coordinates, or complicate the calculations if inner constraints or S-transformations are used, q.v. section 3.3.1.

- First-Order Design (FOD): the configuration problem
  choosing the optimal imaging configuration that will minimise the trace of the a posteriori estimation of the object point coordinate covariance matrix, for a given target array.
- Second-Order Design (SOD): the weight problem
  selecting the required image measurement accuracy.
- Third-Order Design (TOD): the densification problem
  increasing the number of control points beyond a certain number does not affect the precision of strongly convergent close-range photogrammetric networks; this problem is essentially solved during the FOD stage (Fraser, 1983; Dall'Asta et al., 2015; Barazzetti, 2017). It may still be useful for weak imaging configurations or self-calibration, in order to cover the full extend of the frame with observations, but may still be irrelevant for target-less projects as there can be little control over the feature point correspondences.

design strategies

If the scene is unknown or too complicated, an ad hoc imaging configuration can be designed on site based on rules of thumb. However, if the design has not been previously tested, the survey may fail to achieve the quality requirements, unless the network is over-designed (Mason, 1994, p. 21); an approach that may require increased time and resource allocation. Generalising on this notion, in order to guarantee complete coverage and observation redundancy for any given scene, Wenzel (2013) proposed a greedy approach they called "one panorama each step". They claimed that just because high resolution digital cameras and automatic algorithms are available, there is no reason for minimising the number of images or finding an optimal configuration. However, this approach is by no means sustainable considering the implications with respect to the increased memory requirements for large scenes, cf. section 3.3.2 and Hartmann et al. (2016).

The most challenging among the four stages is solving the FOD problem as the solution space is practically infinite. Direct analytical approaches have been proved to be insufficient due to the assumption that the initial solution is close to the optimal and the facts that the cofactor matrix is relatively defined—only random errors are considered—and scene-based restrictions with respect to visibility and camera positions are often hard to implement (Mason, 1995; Marshall, 1989, pp. 81–82). The indirect trial-and-error approach based on computer simulations is the most commonly used strategy. Realistic simulations are not effortless but they can be facilitated by computer-aided design (CAD) systems for the representation of the target array and the camera stations geometry (Luhmann et al., 2006). Monte Carlo simulations have also been used to generate thousands of randomised solutions, by treating each parameter as a statistical value with its own empirical distribution, and selecting the best one. However, the exact adjustments that can improve a given

design simulations

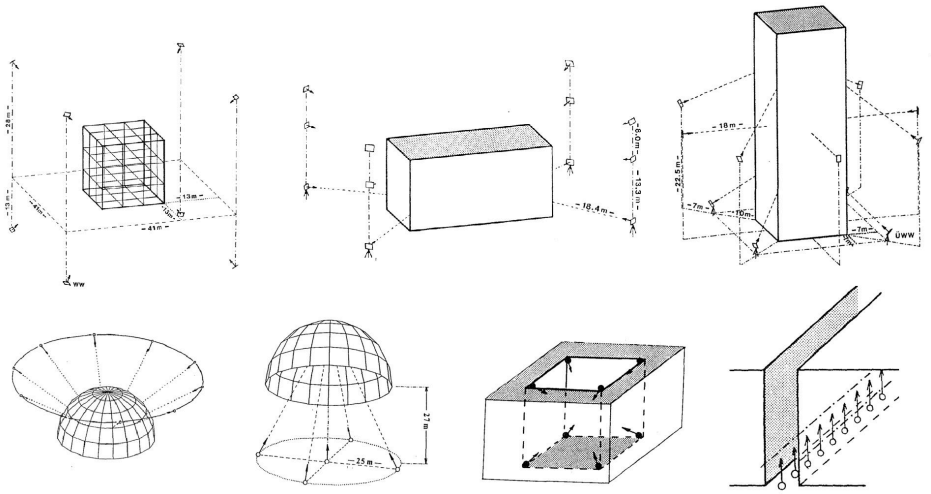


Figure 3.8: Network design simulations on objects of standard geometry; reproduced from Schlogelhofer (1989, fig.6-15,7-17,7-19,7-27,7-37,7-46,7-57).

design are not always obvious and on-site conditions cannot always be realistically represented in an economical way. Therefore, the design process mainly relies on human interaction (Mason, 1994; Fraser, 1996; Hastedt, 2004; Luhmann et al., 2014). This is the main reason why artificial intelligence techniques such as expert systems (Mason et al., 1992; Mason, 1994, 1995, 1997), fuzzy inference systems (Saadatseresht et al., 2005) and genetic algorithms (Olague et al., 2007) have not been successful in practice—apart from their use in next-best-view planning (Tarabanis et al., 1995; Scott et al., 2003; Mostegel et al., 2016) that also applies to active systems and are primarily focused on reconstruction completeness in real time—as there is no simple way, yet, of integrating human experience and expertise in a computer system, beyond a few rules of thumb (Fraser, 1996, p. 278; Triggs et al., 1999, p. 349).

An attempt to formalise such rules, for a few standard cases commonly encountered in urban scenes, was made by Schlogelhofer (1989) by performing simulations on objects of standard geometry, q.v. figure 3.8 and Kraus (1997). Mason (1994, 1995) generalised this idea of standardised imaging configurations by introducing the concept of a generic network; a network that is ideal for the measurement of near-planar surfaces. He designed an expert system that would decompose the scene, optimise several instances of the generic network, and connect them together by adding additional camera stations. Hosseininaveh Ahmadabadian et al. (2013) developed a system that could guide a robot arm carrying a stereo camera rig in creating an elliptical imaging block to completely and optimally reconstruct small artefacts (Hosseininaveh Ahmadabadian et al., 2014). Heikkinen (2005) investigated the potential of the circular imaging block for the reconstruction of indoor environments by simulating a camera fixed on a rotating bar.

Alsadik et al. (2013) developed a network design tool for cultural heritage documentation by amateur users. They initialised the object geometry using a sparse reconstruction based on a video stream, thus, solving the practical problem of simulating a target-less photogrammetric project by creating a 3D model of the scene (Alsadik, 2014; Alsadik et al., 2014). A dense imaging configuration with a canonical shape—e.g. a rectangular block for a building façade or a circular one for a statue—with overlap up to 90% was subsequently filtered and optimised based on criteria related to observation redundancy, object coverage, and position estimation uncertainty. Images of the required views were rendered from the reconstructed 3D model in order to be used as a guide to the user for the on-site image acquisition. Dall'Asta et al. (2015) and Hoppe et al. (2012) used real cameras with miniature objects for laboratory tests. However, even if the realism of such a small camera-object distance can be considered adequate, the size of the miniature of a narrow space that can accommodate a DSLR camera can render this approach impractical.

Not long before the completion of this thesis, Slocum et al. (2017) suggested that rules of thumb established over many years of aerial surveys are inadequate and that empirical assessments which involve on-the-ground reference measurements are too expensive and not repeatable. Taking the idea of Alsadik et al. (2013) a step further, they recreated the virtual 3D scene of a site, thus allowing control over the cameras' interior and exterior orientation, objects texture, and ground control target placement, in order to assess the accuracy potential of UAV-base photogrammetric surveys under different site-specific conditions. To this end, they created CAD models of artificial targets and a simulated scene that represented a 200×200 m relatively flat perfectly Lambertian surface. Two different textures were blended together and applied on the surface. The first one was a 7200×7200 pixel orthoimage, whose footprint in reality covers an 18×18 km area. The second one was a 3456×3456 pixel image of a grass patch that was tiled to create a repetitive 10×10 tile. The assessment of their workflow was based on the surface reconstruction created by dense matching of images of the scene that were rendered using Blender (1995) and were subsequently distorted and contaminated by Gaussian noise. Their premise was that weather conditions, surface texture, camera noise, solar illumination angle, etc. could be realistically simulated within a CAD environment; however, they acknowledged that the graphics engine of the software that was used was not capable of doing so.

3.5 Segmentation and Recognition

One of the main advantages of laser and image scanning methods is that they allow for the collection of a huge amount of information within a limited time frame; as a result, the acquired

geometry includes unwanted measurements. The problem of separating the different objects in a scene from the surrounding environment using segmentation techniques or directly identifying a specific object type in 3D object space and image space is discussed in this section. The aim is to be able to use optimisation techniques for geometry fitting based on the least squares principle, which would otherwise struggle in the presence of observations that do not fit the selected mathematical model; this subject will be discussed in more detail in chapter 4.

3.5.1 Image Segmentation

In sections 3.3.2 and 3.3.3, the full extent of each image was used in order to extract distinctive feature points and depth information. Sometimes, ignoring a portion of an image or focusing on a specific part is beneficial. For instance, if a water body or any other glossy surface is depicted on the images, any feature points detected on it may affect the integrity of the feature matching process. Conversely, one might want to speed-up the dense image matching process by calculating the depth map only for the part of the image where the object of interest is projected.

image segmentation

The straightforward approach to image segmentation involves intensity-based partitioning of the image, the simplest one being thresholding on a specific pre-defined intensity value range in order to create a binary image that defines the foreground and background (De et al., 2016). If colour images are used, Red-Green-Blue (RGB) values are usually converted to Hue-Saturation-Value (HSV) colour space which is more robust against illumination changes (Son et al., 2014; Dimitrov et al., 2014); the saturation channel in particular is considered to be a relative measure of "non-whiteness" (Smith, 1978). Bradley et al. (2007) proposed a method for adaptive image thresholding (Wellner, 1993) which defined a locally estimated threshold instead of using a global value for the entire image. As the method makes the implicit assumption that the image contains primarily background pixels and the foreground pixels are distributed in the image—e.g. text of a book page—a user-defined weight could modify the method's sensitivity ($adThS \in [0, 1]$) towards returning more or less foreground pixels by setting a higher or lower value, respectively.

The premise of segmentation is that each object appears as an area with consistent intensity; by estimating the boundaries of these areas, the silhouette of distinct objects could be extracted and, subsequently, the object of interest can be identified based on both intensity and shape properties. However, this approach is not always successful, as some surfaces of an object may have different colours or textures (Uijlings et al., 2013). Instead of indirectly extracting the boundary of an object, based on assumptions about the homogeneity of its colour, edge detection has been used to directly extract boundary segments assuming high contrast around its periphery; however, joining

together the individual segments has been proved to be very challenging due to missing parts and clutter. Even if a shape prior was used, occlusions and all potential deformations had to be taken into account, as well (Prince, 2012).

material recognition

In an even more direct way of detecting objects in an image, under the assumption that each entity is made of a single material, the segmentation and identification process can be combined in a single step of material recognition. This is essentially a classification problem that requires all the varying visible properties of a material to be compiled to a unique signature. The recognition of materials commonly used in construction of infrastructure works—including sliding tracks—such as concrete or wood, q.v. figure 3.4, is important in applications like as-built modelling and progress monitoring (Dimitrov et al., 2014, p. 38). Generic methods which were developed in other fields may not yield satisfying results on materials such as these, because of their low variability, high similarity, and the fact that objects made of the same material may have different color, shape, structure, or texture (Brilakis et al., 2008; Brilakis et al., 2010, p. 457; Dimitrov et al., 2014, p. 40; Liu et al., 2010). The main problem is that even for objects with rich surface texture, recognising materials in real-world images is affected by object geometry and lighting conditions (Bell et al., 2015).

Neto et al. (2002) observed that in any single image, the color-space values of construction materials occur within a range with canonical boundaries, which had to be defined on an ad hoc basis. Automating the process of identifying these boundaries, even in cases where they are of more complex nature, has been based on extracting features from image databases for each material category. These features have been extracted by clustering raw color values from image patches, e.g. intensity or RGB colour values, or by creating histograms of the responses of the application of several image filters aiming to code texture information (Brilakis et al., 2005, 2008; Zhu et al., 2010). Dimitrov et al. (2014) used a one-class support vector machine (SVM) classifier with non-linear kernels based on colour information, encoded in Hue-Saturation-Intensity (HSI) colour space, and texture encoded in filter responses (Leung et al., 2001). Son et al. (2014) claimed that a correct classification of the aforementioned materials can be acquired using color information alone, based on an ensemble classifier comprising an SVM, a decision tree, and a k-nearest neighbours classifier (Aggarwal, 2015). The training and test datasets comprised 25×25 and 50×50 pixel patches that were manually classified as foreground or background. However, almost half of the collected patches were mixed—meaning that they comprised of different materials and included object edges—and were ignored, which means that the boundaries of an object made of a specific material could not be accurately identified. The same applies to Dimi-

trov et al. (2014) as the segmentation problem was not addressed. In both studies, classification accuracy and precision in the range 80-95% was reported, using medium size patches of approximately 30×30 pixels, which, according to Dimitrov et al. (2014), can go up to 97% for 200×200 pixels. Rashidi et al. (2015) reports similar precision levels from a variety of tested classifiers using similar features, but the recall can be as low as 50%.

Unlike other visual recognition tasks in computer vision, it is difficult to select and fine-tune features that can sufficiently describe each material category (Liu et al., 2010). For example, the necessity of filter banks for texture classification has been questioned, as results from using raw intensity patches as small as 3x3 pixels can be equally accurate (Varma et al., 2003, 2009). The state-of-the-art in visual recognition and image classification are data-driven approaches, such as convolutional neural networks (CNN), that rely on huge training data sets and require significant computational effort during the training stage. A potential solution, known as transfer learning, was proposed by Donahue et al. (2014) and Sharif Razavian et al. (2014); they suggested that using the features identified by a CNN, which has been trained on a vast amount of training data for a particular identification task, can be combined with a simple classifier, e.g. a linear SVM, and have similar representation power in a completely different task compared to finely-tuned features. This means that the automatically learned features could be used for the classification of objects different from the ones they were originally trained for. Nevertheless, no conclusive results exist for BIM-related material categories.

interactive graph cut

In practice, when accurate results are required, user-interaction is necessary. A large family of image segmentation techniques that have showed good results in applications where the aim was to separate an object in the foreground from the background, are based on graph cuts (Shi et al., 2000). In Boykov et al. (2001) each pixel in an image is represented by a graph node. Each edge connecting a node to those corresponding to neighbouring pixels is assigned a cost, which is related to the intensity value and contrast, thus combining colour and edge information. The aim is to find a cut—a subset of the edges—that segments the graph nodes which correspond to a coherent subset of pixels and represent the object of interest and separates it from the background. This is a combinatorial optimisation problem that can be solved in polynomial time. The user has to initialise the solution by selecting seed pixels that belong to the foreground and background. Several extensions and modifications of this algorithm have been proposed, q.v. figure 3.9b. In Rother et al. (2004), the optimisation was performed iteratively in order to improve the result using better initialisations. Li et al. (2004) proposed an algorithm called "Lazy Snapping" that is applied to an over-segmented image, instead of creating the graph using the raw pixels. They



(a) SLIC superpixels, adapted from Achanta et al. (2012, p. 2275;fig. 1).

(b) Image segmentation using Graph Cut (Boykov et al., 2001) and GrabCut; adapted from Rother et al. (2004, p. 311,fig. 2).

Figure 3.9: SLIC superpixels and graph-based image segmentation.

initialised the process with the watershed algorithm Vincent et al. (1991) but superpixels (Stutz et al., 2017) can also be used for the over-segmentation (Ren et al., 2003; Wang et al., 2012). A superpixel implementation called SLIC was proposed in Achanta et al. (2012), q.v. figure 3.9a. This was essentially a clustering process that aims to create superpixels of equal size; the segmentation starts with a regular grid and is optimised based on the perceptually uniform color space CIE (1976) values of the neighbouring pixels. The user needs only define the required number of superpixels and a compactness weight which defines the relative importance between spatial proximity and color similarity when assigning a pixel to a cluster.

The main drawback of all the aforementioned image segmentation techniques is that significant effort is required, either in the form of user-interaction or for training a classification algorithm with a large, relevant, manually classified dataset without any guarantee on the result. Attempts to combine pre-existing or additional knowledge, if available, have led to several hybrid methods. Degol et al. (2016) used images and 3D information, such as depth maps and surface normals, extracted from image-based scanning to material recognition. They suggested that this approach can improve the classification results compared to the method proposed in Dimitrov et al. (2014), which only performed well with standard datasets but struggled with natural scenes. Brostow et al. (2008) used insight from the sparse point cloud created from feature matching on a video stream recorder by a camera mounted on a moving car. By back-projecting points on the images, they were able to transfer knowledge to the image space and enhance the visual cues in order to classify objects in an urban road scene such as road surfaces, buildings, cars, etc. They were able to improve the classifications results from colour and texture information by adding to each pixel

information about the height above the camera, distance from the camera path, surface orientation, the density at which a detected feature was tracked along the frame sequence and its reprojection residual. According to Ladicky et al. (2012) and Hane et al. (2013) the next step towards the integration of image and geometric data sources and processing methods, and arguably a challenging one (Gruen, 2012, p. 51), is the joint estimation of dense reconstruction and object recognition.

3.5.2 Three-Dimensional Shape Detection

Similarly to the discussion on segmentation and detection in image space, q.v. section 3.5.1, these tasks can be either combined or performed separately in 3D space, as well. They can be encountered in a strictly geometric setting, if only the geometry of a scene in the form of an unstructured point cloud is available without any intensity information, q.v. section 3.6.2. Currently, as-built 3D modelling is a manual or semi-automated process and commercially available software require user interaction (Son et al., 2015, p. 180; Dimitrov et al., 2016, p. 483); the reconstruction of a typical piece of infrastructure can potentially require weeks or months of tedious work (Tang et al., 2010, p. 831).

As free-form objects are more complex (Campbell et al., 2001; Pu et al., 2009), the attempts to automate aspects of the as-built modelling process for BIM-specific projects are mainly concerned with the detection and modelling of primitive objects that have simple geometry—i.e. planes, spheres, and cylinders— (Tang et al., 2010; Xiong et al., 2013a; Volk et al., 2014; Bosché et al., 2015; Patraucean et al., 2015; Son et al., 2015) and involve two main approaches (Tang et al., 2010). The first as-built modelling approach is based on extracting cross-sections and extruding them to create the desired surfaces. The cross-sections may also be converted to binary raster images in order to use image segmentation and template fitting methods (Jung et al., 2014). This essentially transforms the problem to a 2D shape recognition and fitting process, q.v. section 4.3. The second one involves fitting geometric primitives to point cloud subsets. The point cloud can be segmented based on generic criteria based on local geometry, such as density, curvature, roughness, or normal direction. Rabbani et al. (2006) initially segmented the point cloud using smoothness constraints based on surface normals. A similar region growing segmentation technique based on surface curvature and roughness was proposed in Dimitrov et al. (2015) for the detection of planes and cylinders in buildings and industrial sites. However, as Sareen et al. (2011) pointed out, algorithms based solely on surface curvature may fail to delineate objects with similar curvature properties or when surface boundaries are occluded. In addition, the threshold selection for density or roughness criteria requires knowledge of the noise properties of the scanning result which

point cloud segmentation

may be hard to infer, control, or generalise.

prior knowledge

In Weinmann (2016), a framework was proposed that attempts to assign a semantic label to each point in an urban road scene point cloud. The points are classified to categories such as ground, façade, vegetation, or car based on a selection of 2D and 3D features extracted in their vicinity. The effects of the neighbourhood size, classification algorithm, and feature combinations were also assessed but the amount of noise, clutter, and outliers in the point clouds under consideration was insignificant. This approach can be regarded as the first step in a hierarchical scheme where prior knowledge about each object can be used for the subsequent fitting of specific geometry. Such prior knowledge about the scene in the form of as-designed CAD models was used in Bosché et al. (2015) and Czerniawski et al. (2016) in order to detect and fit cylinders after locating the specific pipe components in a cluttered point clouds.

shape recognition

Alternatively the object recognition task can target an entire geometric class, e.g. planes or cylinders, or a particular instance of it, e.g. a sphere with 2 m radius (Tang et al., 2010). In Vosselman et al. (2004) and Schnabel et al. (2007) planes, spheres, and cylinders are directly detected in laser scanning point clouds using a 3D Hough transform (Hough, 1962; Ballard, 1981) and the RANSAC algorithm (Fischler et al., 1981), respectively. Pu et al. (2009) initially detected planar surfaces in laser scans of building façades that were subsequently assigned semantic information based on their size, position, orientation, topology, and point density. Flat surfaces are the primary interest in creating BIM model of building interiors where contextual information is also used—e.g. walls are vertical and perpendicular to the floor, doors and windows are openings on walls (Budroni et al., 2010; Thomson et al., 2015). The extracted surfaces are either used to assist manual modelling (Jung et al., 2014) or attempt to eliminate user interaction (Xiong et al., 2013a).

contextual information

3.6 State-of-the-Art Software and Hardware Implementations

The theoretical aspects of the fundamental processes in photogrammetric vision have been well established for quite some time now; nevertheless, the software and hardware tools for 3D data acquisition and shape modelling will keep evolving. Thus, only a snapshot of what is considered to be the state-of-the-art in image-based and hand-held laser scanning at this point in time is presented in sections 3.6.1 and 3.6.2, respectively.

3.6.1 Image-Based 3D Scanning Software

A plethora of digital photogrammetric workstations for the processing of aerial images has been developed during the last three decades. These are expensive, often modular, software solutions

for topographic mapping from regular image blocks (Granshaw et al., 2015, p. 6; Forlani et al., 2015, p. 87). On the other hand, several free—e.g. the now discontinued Photosynth (2008) and Autodesk 123D Catch (2009)—or even open source applications, such as Bundler (2008) and Apero (Pierrot-Deseilligny et al., 2011), started appearing almost ten years ago. They were able to automatically process images that were acquired with highly irregular configurations and, in combination with a dense image matching algorithm, produce dense point clouds. There are now quite a few software packages available for automatic image-based 3D modelling and several of them were considered for this project. As a full-scale comparison between them was beyond the scope of this work, they were only evaluated with respect to their user-friendliness and suitability for track scanning. The objective was to identify a stable workbench in order to be able to focus on the application at hand.

open source

Most of the open-source solutions were soon excluded from further consideration due to practical and performance reasons. On the one hand, the Apero-MicMac package maybe the most flexible option based on a solid scientific foundation used in many photogrammetric research projects, but most resources are in French and until recently it lacked a graphical user interface (GUI). On the other hand, VisualSFM (Wu, 2011) is a GUI environment that implements an SfM workflow, exploits the parallel computation capabilities of the central (CPU) and graphics processing units (GPU), and can be combined with several dense image matching implementations such as PMVS or SURE. However, it is not suitable for forward camera motion, according to its documentation and a few simple tests, because it only uses features detected at higher image scales (Szeliski, 2011, pp. 116–117; Klette, 2014, p. 57)—for tie-point matching (Hartmann et al., 2016, p. 51).

commercial packages

Four commercially available software packages were also considered. Pix4D (2017) is primarily targeted to, and is perhaps the most popular for, aerial mapping applications using UAVs. RealityCapture (2016), even though it has remarkable processing capabilities, it appeared to have very strict quality requirements for the matching process. PhotoModeler Scanner (2016) is primarily used in high accuracy close-range engineering applications but did not perform well in simple tests with the data sets under consideration and was quite slow in processing larger data sets. PhotoScan (2018) is a very popular option with a wide user-base and is a relatively inexpensive, as its standard version costs less than a couple of hundred dollars. It is also the most user-friendly option that not only includes automated processes but can also be upgraded to a version that provides a Python (1991) application user interface (API) that can be used to further automate customised workflows. More importantly, even though it does not output a posteriori precision estimations, after the initial SfM stage it follows a traditional photogramemtric workflow and has been exten-

sively used by the scientific community and good results were reported using external reference data (Remondino et al., 2014).

DBAT toolbox

An initiative led by Borlin et al. (2016) developed an independent toolbox—called DBAT—for the external verification of PhotoScan's computations. It could also be used to extend PhotoScan's capabilities by further processing its output using different sensor models and calculating the covariance matrices for the estimated object points. It is reported therein that a significant discrepancy was identified in PhotoScan's bundle adjustment results compared to PhotoModeler which was attributed to the way the ground control points are used. However, as we pointed out (GitHub DBAT Issue, 2016), this was due to a bug in the original contribution that was caused during data conversion. This was fixed in version 0.6.2.2 but further testing by the community is required before the latest DBAT toolbox version can be considered stable. In a previous assessment, Remondino et al. (2012, p. 52) expressed their surprise upon discovering that despite the unstable interior orientation estimates for non-metric cameras from low-cost and open-source image orientation software, including PhotoScan, the object space accuracy of a self-calibrating bundle adjustment can be more than acceptable. Even though they presumed that the exterior orientation compensation is to blame, they suggested that further investigations are required; however, this is a well-known effect of projective coupling, q.v. section 3.3.1. Remondino et al. (2012, p. 47) also attributed the potential uncertainties of, the so-called, SfM tools and methods to the image variant interior parameter set. Nevertheless, there is no SfM manifesto written in stone; this is a choice with regard to the functional model that is either justified by the camera stability and sustained by the imaging configuration or not. It all comes down to the robustness of the SfM scheme with respect to the detection of outliers.

criticism

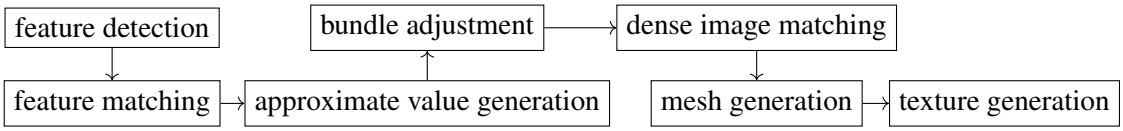


Figure 3.10: The typical image-based 3D modelling workflow in Agisoft PhotoScan.

3.6.1.1 PhotoScan

As ©Agisoft PhotoScan (2018) was used for some of the experiments which are presented in this thesis, a brief overview of the workflow that is followed within the software is provided, in order to highlight certain parameters that may affect the otherwise automated process.[2] It is worth noting,

[2]Agisoft PhotoScan Pro version 1.4 was installed on a laptop running on 64-bit Microsoft©Windows 7 Enterprise SP1, Intel®Core™i7-4800MQ CPU@2.70GHz, 32GB RAM, NVIDIA Quadro®K2100M.

however, that the exact details of its proprietary algorithms are not disclosed in the documentation. After importing the images, a quality metric can be calculated for each one based on the relative sharpness—in a 0–1 range—of the most focused part of the image with respect to the sharpest image in the data set. Setting a quality threshold, even a relative one, is not straightforward when no reference image is used as its value is related to the content of each image (Charrier et al., 2015). The recommended minimum value is 0.5, but striking a balance between keeping only the sharp images and at the same time the required number of images, in order to maintain sufficient overlap between successive images along a sequence, is often hard (Rashidi et al., 2013, p. 273).

Five processing settings are available for the image orientation process; at the highest level, the image resolution is upsampled by a factor of four, thus doubling their size, in order to increase the number of detected points and their localisation accuracy—this setting is recommended for sharp images only and is potentially the most time consuming. At the high level, the original images are used and for each one of the last three settings their resolution is reduced by a factor of four.

Limiting the search space for feature matching can be achieved by setting an upper limit to the number of detected points—the recommended maximum number of match points is 4000—and providing approximate coordinates for the camera exterior orientation parameters. If the camera is not precalibrated, an adaptive camera model can be used during the local, incremental bundle adjustments for the approximate value generation; one which extends the standard lens distortion model, q.v. section 3.3.1 by using more coefficients for the lens distortion and additional parameters to account for shear and affinity, in order to prevent the divergence of the standard model parameters when weak imaging configurations are considered.

Uncertainty estimations can be assigned to the manually observed or automatically detected image coordinates, ground control point coordinates, and exterior orientation parameters in order to weigh their effect on the optimised metric during the bundle adjustments but no a-posteriori covariance matrices are calculated. Potential outliers among the object points can be removed from subsequent adjustments based on their "reconstruction uncertainty" (RU), which increases with short baselines and small convergence angles, and three additional parameters related to their corresponding image projections:

- the total number of image projections
- the a priori localisation uncertainty, called "projection accuracy" (PA), which corresponds to the image scale on which they were detected
- the total reprojection error (RE)—in pixels, scaled according to the corresponding projection

accuracy—which represents the a posteriori localisation accuracy taking into account all image projections (PhotoScan, 2018).

Similar quality levels are provided for dense image matching—according to Remondino et al. (2014), the implemented algorithm is based on the semi-global matching algorithm (Hirschmuller, 2005)—starting at full image resolution, combined with four filtering options for the depth maps: no filtering, mild, moderate, and aggressive. The area of interest should be surrounded by a bounding box in order to restrict the disparity range and the depth maps and reduce the size of the generated point cloud and mesh.

Two methods for surface reconstruction are provided. The first one requires a dense point cloud as input and, thus, can be used with point clouds from any source, e.g. laser scanner. It is presumed to be an implementation of Poisson surface reconstruction based on Kazhdan et al. (2013) which creates watertight closed surfaces; its use for open surface reconstruction means that small holes are filled but a criterion needs to be defined in order to remove the triangles that are not part of the surface. It appears that the size of the bounding box implicitly defines several other parameters of the surface reconstruction algorithm as the only customisable user-defined parameter is the maximum number of triangle faces in the output mesh; the recommended value is one face for every five points in the dense point cloud. The second mesh generation method, which is called "visibility consistent", creates a mesh directly from the depth maps without the intermediate phase of point cloud generation in order to avoid taking into account noisy points—as it was experimentally introduced in the software towards the end of this study, it is only briefly assessed in section 11.2.

Image masks can also be used in two ways: either to only consider a part of the image for the disparity and depth estimation that will be used for the dense point cloud generation or to strictly force any ray from the projection centre through the pixels that have been marked as background to pass through empty space. The former masking option may still result in noisy point clouds as a result of the depth map fusion process. The latter, strict masks, are more appropriate for small objects that have been imaged in their entirety, and therefore not considered in this study, as any part of the object that is projected on the background will be ignored, even if it is not visible on the masked image. Consider for example the images in figure 3.6; if the blinds are correctly marked as background on those images, the part of the track behind them will be ignored in the reconstruction even if it is visible on the images farther down the track.



Figure 3.11: The Zeb1 hand-held laser scanner.

3.6.2 Hand-Held Lasers Scanners

Zeb1

Not long before the beginning of this project, a new generation of hand-held laser scanning devices was introduced with a potential in survey applications, namely the Zeb1 (Bosse et al., 2012). The Zeb1 comprises a ©Hokuyo (2015) UTM-30LX laser scanner combined with a ©Microstrain (2015) 3DM-GX2 inertial measurement unit attached to a spring and a data logger attached to the power source, q.v. figure 3.11. The spring has a dominant resonant pitch and roll frequency of 1.4 Hz (Bosse et al., 2012, p. 1106), which correlates well with the average human walking pace, and has to be constantly moving during the scanning process (Danion et al., 2003; Williams, 2014, p. 133). The beam of the laser unit oscillates on a plane providing a 270° field of view while the sweeping motion of the spring allows for up to 30 m range measurements—15 m outdoors—at a 3D field of view. The collected distances are segmented at packets of approximately one second duration during which the sensor's position and orientation can be adequately estimated from the IMU measurements. The 3D features that are identified in each individual point cloud are iteratively matched together and used to estimate an optimal transformation between them, similarly to the the iterative closest point (ICP) algorithm (Chen et al., 1992), in order to build up the full scene. Finally, if places that have been revisited are identified, a loop closure optimisation process is performed aiming to reduce the inevitable position estimation drift. For more details see (Bosse et al., 2008, 2009, 2012, 2013; Williams, 2014).

web-based processing service

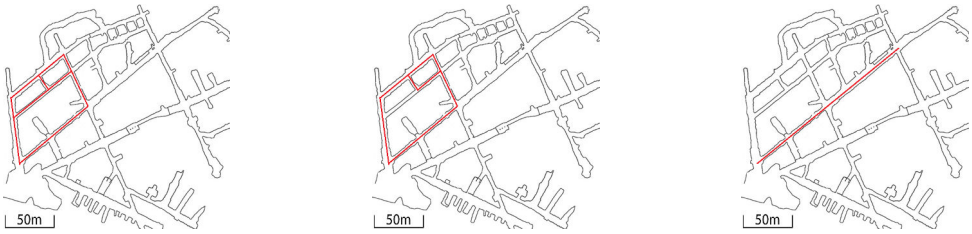
Post-processing of the data set is conducted automatically on a web-based processing service and lasts as long as it takes to collect the data. If the automatic process fails to identify a loop where it exists—due to the distance between corresponding points in the initial solution being, typically, greater that 10m—the user can flag the data set so that it can be manually processed. The output is the sensor's trajectory and a point cloud of the scene, color-coded according to acquisition time and position estimation uncertainty. No information about intensity, let alone RGB colour, is available due to bandwidth limitations that could hinder data synchronisation and negatively affect the data

fusion between the laser scanner and the inertial measurement unit. Point clouds shaded and colour-coded according to the distance of each point from the sensor trajectory are also provided. The typical 3D accuracy is defined as 0.1% of the size of the scanned object; the recommendation is to walk at a normal pace—which can be conservatively interpreted as approximately 0.7–1 m/s or 2.5–3.5 km/h—for no longer than 20 min (GeoSLAM, 2015). Any reference points in the scene are only used for a rigid similarity transformation in order to georeference the point cloud, which is already scaled with the Z axis oriented along the gravity vector measured by the IMU (Williams, 2014, pp. 154,169,172,213).

underground trials

The results of theoretical and practical investigations with respect to the data acquisition rate and the nominal and achievable accuracy of the Zeb1 system are summarised below. It should be noted that the average speed was not explicitly specified in any of the work that was reviewed; therefore, the reported walking speeds were estimated based on the available data. Williams (2014, p. 149) reports that the theoretical positional accuracy at 10 m range from the sensor with respect to operation time follows the equation 0.045 m + 0.013 m/min. Three of the data sets presented therein, which were acquired along mine tunnels at an average walking pace of 1900 m/h, are depicted in figure 3.12. A 170 m long tunnel was scanned within 11 minutes by walking from the beginning to the end and back from the same path—a total length of 340 m—q.v. figure 3.12c. The result was a 10.3 m distance between corresponding points at the beginning of the tunnel scanned during the first and last minutes of the survey. The loop closure did not improve the result because corresponding points were detected all along the tunnel creating small loops that failed to propagate the corrections along the full length of the sensor trajectory. The initial open loop solution for the path depicted in figure 3.12b, that comprised a large and a smaller loop, had approximately the same length, and was also scanned within 11 minutes returned a closure error of 12.5 m. After being flagged as erroneous the error was reduced to an acceptable level.

Three possible explanations were considered for the large errors that prevented the loop closure detection. Firstly, the geometry of the scene did include acceptable 3D features as the rest of the scans were successful, so it was ruled out. Secondly the walking pace was considered and was deemed slow enough during both trials—from the lengths and scan durations it was estimated to be in the range of 1900–2800 m/h. Conclusively, the errors were attributed to the presence of another person walking behind the sensor at a similar pace. Ignoring the first data set, the reported accuracy from a total of 4 scans along the tunnels depicted in figure 3.12a, compared to a reference point cloud acquired with a terrestrial laser scanner, was reported to be on average at decimetre level even though it could locally reach 2.25 m.



(a) Test site A. **(b)** Test site B. **(c)** Test site C.

Figure 3.12: Top view of the tunnels at the Camborne School of Mines test site, adapted from Eyre et al. (2016, fig. 2).

The path in figure 3.12a, at same mine test site, was also surveyed in Eyre et al. (2016) using the Zeb1. The survey lasted twice as long compared to that reported in (Williams, 2014), as certain segments were scanned more than once. This is probably the reason why the results reported therein, compared to a reference point cloud produced by a terrestrial laser scanner, are even more optimistic, setting the overall point cloud accuracy equal to the nominal range measurement accuracy of 3 cm. As the exact lengths of the sensor trajectories are not specified, based on the sensor trajectory illustrations (Eyre et al., 2016, fig. 3) and the indication that each scan lasted between 20 to 30 minutes, the walking pace was estimated at approximately 1000–1200 m/h. In addition to the Zeb1, two more sensors were tested in Eyre et al. (2016). A prototype of the next generation of the Zeb laser scanner and the ZEB-REVO that was released in 2016. The spring was replaced by a rotating mechanism with a 0.5 Hz frequency so the user simply walks around while holding the sensor without rocking it. The rotational frequency of the laser unit was also upgraded from 40 Hz to 100 Hz. A camera can be attached to the handle but the points still have no intensity or colour values assigned. The captured images are stitched together to create a 3D panorama that is synchronised with the point cloud within a dedicated viewer in order to help the user identify structures in the captured geometry. A desktop software application is also available for local processing of the laser scans.

indoor trials

In Thomson (2016) the Zeb1 was used to scan a $39 \times 7 \times 5m$ building corridor within 10 minutes. Assuming that the sensor trajectory follows the perimeter of the corridor, as the exact path that was followed was not defined therein, the walking pace during scanning was 550 m/h. For the reference point cloud survey, which lasted 4 hours, a ©FARO Focus (2017) terrestrial laser scanner was used. The reported accuracy, based on the comparison of door and window widths as well as wall-to-wall distances to their reference values, was found to be on average below 10 cm and at most 50 cm, which is consistent with what was reported in Williams (2014, pp. 179,189) for an indoor trial scan of a 3 story office building interior that lasted lasted 22 minutes.

Chapter Summary

This chapter was focused on all the necessary steps within an automated photogrammetric computer vision system that are required in order to represent a 3D scene, which was captured by a hand-held sensor, and analyse its geometry; the interpretation of the reconstructed scene will be discussed in chapter 4 on geometry processing. More specifically, planning of a photogrammetric project was discussed in the context of a sliding track site in order to identify all the properties that may affect the result, as environments of similar geometry covered in ice are not common in close-range applications. The photogrammetric principles of the image-based track-scanning method and the network design simulations that precede it, as they are implemented in commercial software packages that provided automated SfM workflows, were also discussed. In this context, a brief overview of the bundle adjustment implementation of Agisoft PhotoScan, which relies on the collinearity equations and uses a combination of feature detectors and descriptors for non-signalised scenes for the image orientation and a state-of-the-art algorithm for dense image matching was presented. Given the fact that no coded targets will be used during the image-based scanning survey, the traditional photogrammetric network design processes should be adapted in order to take into account the feature detection process, instead of using rigid assumptions about the geometric distribution of the scene's points within the bundle adjustment.

Even though an offline post-processing workflow is considered in this study, the potential of integrating an online feedback loop should also be considered. However, in order for it to be appropriate for a hand-held sensor within the time constraints for the task at hand, its should ideally be based on image quality metrics whose computational footprint is significantly lighter than existing approaches that require approximately 2 seconds per image on a standard laptop. Methods for image segmentation as a pre-processing step were also discussed. As there is no indication that segmenting the track on the images can be translated to clean 3D mesh of the track, in the first instance—instead of using the latest, more complicated, and computationally demanding state-of-the-art methods for semantic image segmentation or material recognition—a more conservative approach that involves the automation of a simpler, interactive graph-based approach might be more appropriate. Finally, the latest generation of survey-grade hand-held laser scanners was reviewed. Even thought their theoretical accuracy has been confirmed for confided indoor spaces, there is no indications that it can be replicated or extrapolated to longer outdoor corridors.

Chapter 4

Geometry Processing

In chapter 3, aspects of the measurement process using optical techniques were addressed. These techniques can be thought of as discrete sampling processes that provide the input to geometry processing algorithms for the creation of continuous digital representations of real objects, which are discussed in this chapter. Several problems arise from the fact that the observed points are irregularly spaced—the sampling density cannot be influenced directly, as it varies with the surfaces' reflectance properties and depends on occlusions and depth variations caused by the scene's geometry. Unlike the reconstruction of 1D signals—where applying the Nyquist-Shannon theorem (Unser, 1999) can provide a solid basis on how to conduct the sampling process, even for irregular and finite sampling Förstner et al. (2016)—propagating the uncertainty in the detection of image features (Morris et al., 2002) or in the distances measured at single points in order to estimate the precision of the 3D model is not straightforward (Stout et al., 1993) because the geometry generation process can be quite complicated, q.v. figure 4.1. As the generated point clouds will inadvertently include points which were measured on surfaces that may be of no interest for a specific application, the segmentation and detection aspects of geometry processing, which were briefly discussed in section 3.5, need also be addressed.

sampling

Two of the tasks that are involved in the generation of the 3D model of a scanned object are inferring the qualitative and quantitative properties of unobservable quantities and filtering out noise in measurements (Stout et al., 1993; Förstner et al., 2016, pp. 727–735). Both of them can be solved based on assumed models for the geometry of the objects of interest which will be fitted to the available observations. In section 4.2, the subject of curve fitting to noisy and incomplete point sets, which were collected along a roadway centreline alignment, is discussed. Surface fitting techniques relevant to infrastructure works with geometry similar to that of a track are presented in section 4.3 and appropriate geometric shape representation schemes are discussed in section 4.1.

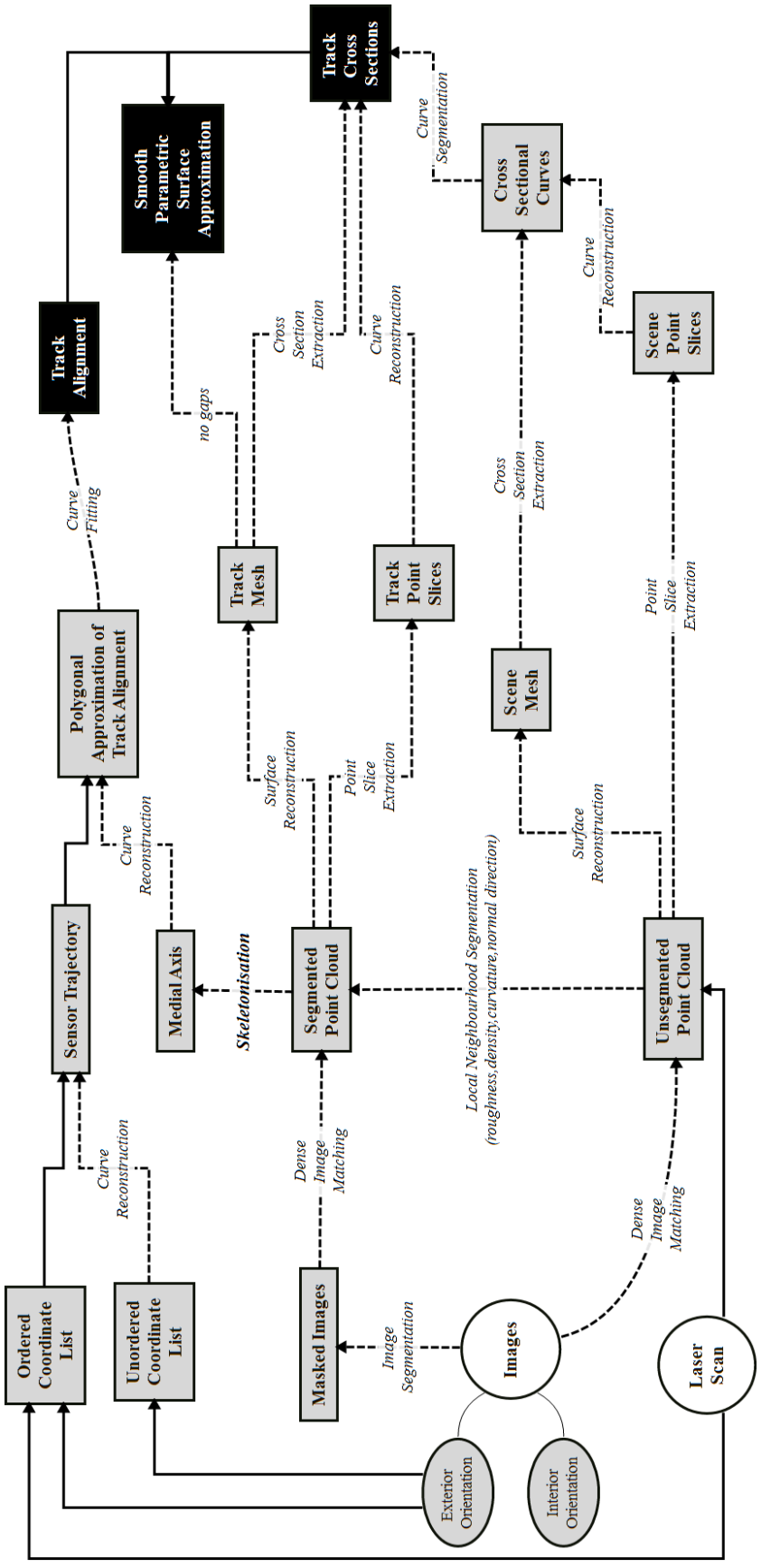


Figure 4.1: Alternative workflows for the 3D reconstruction and modelling of a sliding track's surface.

4.1 Geometric Shape Representation and Optimisation

In order to effectively approximate the geometry of a race track or a road, appropriate mathematical models are required for the representation of curves and surfaces; volumetric representations are not relevant as only the visible surfaces, and not the entire volume, of these objects are of interest. The most straightforward way of constructing a mathematical abstraction of the surface of an object which was measured by a laser or image-based scanning method is by defining the topology of the scanned point set. For this purpose, explicit representations of curves or surfaces are commonly used in the form of polygonal chains or triangle meshes, respectively, which express the points' connectedness; in the context of geometry processing this task is referred to as reconstruction (Dey, 2006).

explicit

An explicit representation is defined by the entire point set, which means that appropriate data structures are required for the efficient manipulation and simplification of large meshes. An important quality of a surface reconstruction algorithm is its ability to avoid creating a non-manifold 3D triangle mesh; that is a mesh that includes edges or vertices such as those depicted in figure 4.2, which may cause problems in further processing (Botsch et al., 2010). The same applies to curve reconstruction techniques where curves with crossings—such as the projection of centreline of a track with a kreisel bend on a satellite image—need extra attention (O'Rourke et al., 1987; Dedieu et al., 1994; Liu et al., 2005; Nguyen et al., 2008; Bo et al., 2016). A variety of techniques that can generate a mesh from a point cloud already exist; the Poisson surface reconstruction algorithm (Kazhdan et al., 2006, 2013) being one of the most popular. However, when it comes to noisy point clouds with gaps, significant user-interaction is required in order to get a desirable outcome (Berger et al., 2016).

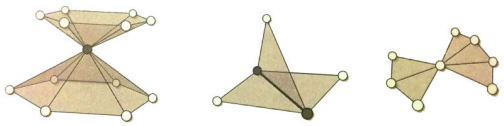


Figure 4.2: Examples of non-manifold triangle mesh vertices and edges; adapted from Botsch et al. (2010, p. 12;fig. 1.6).

implicit

A more compact representation of a point cloud can be achieved by using a single function that implicitly describes the relation between the coordinates of the points lying on its surface. For example, a unit sphere centred at the coordinate system's origin can be defined by the equation $f(x, y, z) = x^2 + y^2 + z^2 - 1 = 0$. However, several surface patches are required in order to describe more complex objects, in which case boundary problems of connectivity

and smoothness arise. Such issues can be handled well by a parametric representation. Using parametric the same example of a sphere, the 3D coordinates of its surface can be described with respect to two independent parameters, $u \in [0, \pi]$ and $v \in [0, 2\pi]$, by explicit functions of the form $S(u, v) = [x(u, v), y(u, v), z(u, v)]^T = [sin(u)cos(v), sin(u)sin(v), cos(u)]^T$.

The parametric representation is not unique for a particular shape; any arbitrary function can be used to describe each coordinate, with the most common one being the family of polynomials expressed in Bézier or power basis form, allowing for the representation of a wide variety of curves. A single polynomial function can be used to describe free-form shapes using a small number of control points; however, certain types of curves and surfaces such as circles, ellipses, spheres, or cylinders, cannot be exactly represented and require rational polynomial functions. Bézier curves can adequately describe complex geometries if more control points are used but the lack of local control can be a problem as a change in one of them affects the entire shape; using a single polynomial of higher order to represent shapes of increasing complexity is not recommended, either. Alternatively, in order to be able to overcome such problems, more segments can be combined to construct an accurate piecewise polynomial or rational representation, e.g. B-splines or NURBS (Piegl et al., 1997).

Each representation has its advantages and disadvantages. Determining whether a point lies on a surface is straightforward with the implicit form whereas with the parametric representation sometimes one has to deal with "parametric anomalies", such as the representation of the poles of a sphere. Conversely, tasks such as representing curve segments and surface patches, traversing a curve in a specified direction in order to generate an ordered point sequence, and the digital representation in a computer system come naturally for the parametric representation. As all the above are complementary tasks serving the same purpose of geometry processing, being able to convert between representation schemes is essential (Campbell et al., 2001, pp. 169–174; Tang et al., 2010, pp. 833–834; Botsch et al., 2010). In line with the interpretation component of the metamodel presented in figure 3.1, the mathematical representation of geometrical features needs to allow for the simple integration of existing knowledge in the geometry prediction process, in order to compensate for missing data and identify outliers. The implications of using the aforementioned methods—which make implicit smoothness assumptions based on the degree of the polynomials that are used in order to construct the curve—on the representation of road and track centreline alignments are discussed further in section 4.2.

The task of curve or surface fitting involves the estimation of the parameter values of the registration selected representation of the geometric model of an object. This process is closely related to the

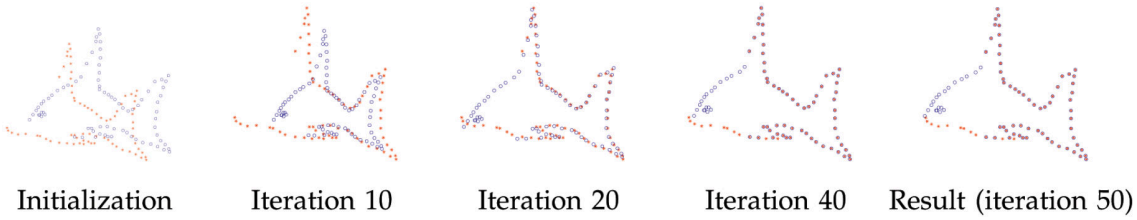


Figure 4.3: A non-rigid curve registration example using the Coherent Point Drift algorithm with $w = 0.5$, $\beta = 2$, and $\lambda = 2$; adapted from Myronenko et al. (2010, fig. 11;p. 2272).

curve or surface registration problem where the geometry of the object is assumed to be known but an unknown, yet controlled, deformation or an optimal transformation—based on the observations, i.e. the scanned points—is sought for. As the track's cross section shape varies reasonably with respect to a canonical shape, q.v. figure 2.4, this approach could potentially override the curve segmentation, reconstruction, and fitting problem. An extensive review of the available rigid and non-rigid registration techniques was given in Tam et al. (2013, p. 8) where the use of templates to simplify the part-to-whole registration problem and effectively handle noise and missing data was pointed out.

One of the most powerful and flexible methods for finding the rigid, non-rigid, or affine registration parameters and correspondences between two 2D or 3D point sets was the "Coherent Point Drift" (CPD) algorithm proposed by Myronenko et al. (2010). CPD formulates the registration process as a probability density estimation problem, where one point set represents the Gaussian mixture model (GMM) centroids, which is moving coherently as a group to preserve its topological structure, and the other one represents the target data points; the blue and red point groups in figure 4.3, respectively. The correspondence probability between two points is defined as the posterior probability—whose likelihood needs to be maximised—of a GMM centroid given the data point. Three user defined parameters w, λ and β are required, as well as a maximum iteration limit. Parameter $w \in (0, 1)$ reflects the assumption on the amount of noise and outliers; λ and β reflect the amount of smoothness regularization. Parameter β defines the width of the smoothing Gaussian filter and controls the locality of spatial smoothness, whereas λ represents the trade-off between the goodness of maximum likelihood fit and regularization.

optimisation

There are many optimisation frameworks that are commonly used in engineering applications, including curve and surface fitting. The bundle adjustment method, for example, combines the solution to the geometry fitting and registration problems in a single framework, q.v. section 3.3.1. The established correspondence between object and image space points allows for a straightforward definition of the objective—or cost—function, which quantifies how good a candidate

solution is; optimising the cost function ϕ implies finding values for the unknown parameters $\boldsymbol{x}$ that minimise the reprojection errors: $\underset{\boldsymbol{x}}{argmin}(\phi(\boldsymbol{x}))$. The nature of the collinearity equations is such that their partial derivative estimation is straightforward, thus, it can be solved by gradient-based methods (Rao, 2009). For problems that require a trade-off between several—often contradicting—criteria where, typically, there does not exist a single solution that simultaneously minimises all objective functions, i.e. $\underset{\boldsymbol{x}}{argmin}(\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \phi_3(\boldsymbol{x})...)$, two strategies are mainly used. One of them is scalarisation, where the multiple criteria are combined in one; the main problem is finding an appropriate weighting scheme that conveys the importance of each criterion. Multi-objective optimisation techniques are also applicable; instead of a single solution, their aim is to return a set of solutions, called the Pareto-optimal front, which cannot be improved with respect to any of the criteria without degrading at least one of the rest; choosing one of them as the final solution is usually left to the designer's preferences. An appealing option for such problems is using genetic multi-objective algorithms. These are global optimization techniques which do not require gradient information and can be effective irrespective of the type of the constraints and objective functions. Instead of a single solution at a time, they evaluate and improve a large number of candidates using random numbers and information from previous iterations (Holland, 1975; Marler et al., 2004; Deb, 2011).

Minimising a cost function for curve or surface fitting to a discrete point set involves the estimation of the closeness of a candidate solution to the input data which in turn requires the computation of the distances between corresponding points in the input and on the candidate curve or surface. The correspondence can be established using the aforementioned parameterisation schemes. The distance that expresses the positional error can be defined as the Euclidean distance between the points or the distance from the input point to the tangent line or plane to the candidate curve or surface, respectively, at the corresponding position. For least-squares curve approximation schemes, where the observed point parameterisation and the endpoints of each polynomial segment are considered to be unknown, geometry fitting is a non-linear optimisation problem (Piegl et al., 1997, pp. 405–419). If the initialisation is close enough, the probability of a gradient-based least-squares method converging to a correct solution increases. The rate of convergence to the correct solution also depends on the distance definition which can be defined as the orthogonal distance from each point to the candidate solution or vice versa (Pottmann et al., 2002; Wang et al., 2006). The parameterisation and correspondence problem can be entirely overridden if an appropriate initial approximation of the correct curve or surface geometry is available.

As no contribution is claimed on the mechanics on any of the well-known optimisation algo-

rithms which are used herein, and assessing any particular implementation is beyond the scope of this work, the reader is referred to the corresponding textbooks, research articles, and ©Mathworks MATLAB (2017) documentation for further details on theory and implementation. The objective functions that were used and the effectiveness of each method in solving the relevant geometric problems are discussed in detail in the relevant sections.

evaluation

Regardless of the method that was used to solve a geometric fitting problem, the result of an optimisation process should be subjected to verification. A straightforward way of evaluating its accuracy is comparing it to a reference object. Alternatively, two different criteria can be used, namely topological accuracy and structural recovery. In the context of curve reconstruction, the former assesses whether the correct number of loops, junctions, or crossings were reconstructed. The latter takes into account the extent to which the expected structural interrelationships, e.g. parallelism, co-planarity, orthogonality, which can potentially improve robustness against missing data, were recovered, preserved, or reinforced. The overall method can also be assessed based on whether or not its results can be easily reproduced between different calculations (Berger et al., 2016, p. 23).

4.2 Curve Fitting

The representation of complex geometry with a combination of several generic polynomials, which can be initialised in their simplest form using a curve reconstruction scheme, was discussed in section 4.1. The design of sliding tracks, however, is based on a specific type of curves called clothoid splines (Meek et al., 1992; SAIT, 2012; Rempfler et al., 2016), q.v. figure 4.4, which comprise straight line segments—henceforth called tangents—and a combination of clothoid segments and circular arcs—referred to as bends. This is considered to be the most appropriate curve model for the representation of road centreline alignments, as well; thus, clothoids have been widely used in road and railway design as transition spirals (Levien, 2008). Their linear—with respect to the arc length u—curvature profile can accommodate the dynamic movement of vehicles along bends as it correlates well with the movement of the steering wheel. Their aesthetically pleasing shape has also attracted attention in sketch-based modelling (Baran et al., 2010) and shape completion (Kimia et al., 2003; Harary et al., 2012). Their role was to maintain a piece-wise linear curvature profile along curve segments where no geometry information was available.

clothoid

The Cartesian coordinates $[X(u)Y(u)]^T$ of an Euler clothoid segment are given by the Fresnel integrals, q.v. equation 4.2, where A is a scale factor. One of the problems that are related to clothoids is the computational complexity in calculating its parameters, which can be addressed

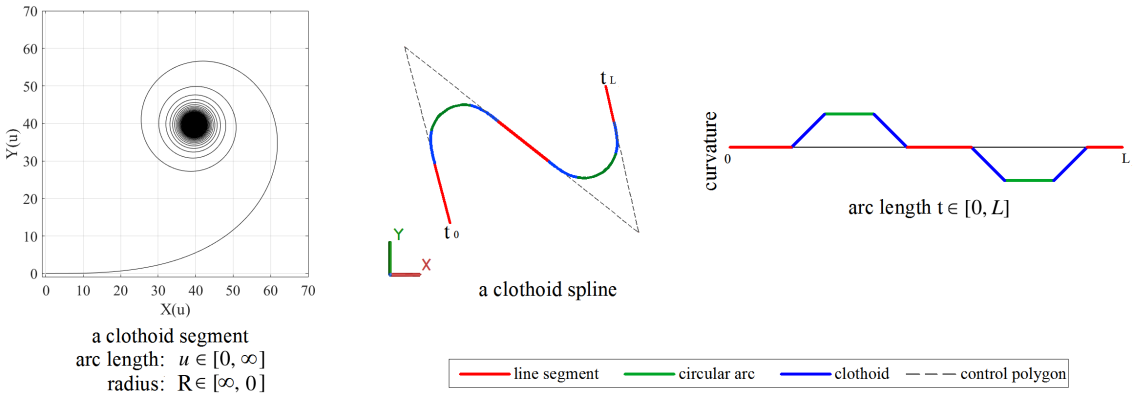


Figure 4.4: Clothoid curves and their curvature profile.

by adequate approximating formulae, q.v. equation 4.3 (Heald, 1985; Bertolazzi et al., 2014; Vázquez-Méndez et al., 2016).

$$\begin{bmatrix} X(u) \\ Y(u) \end{bmatrix} = \pi A \begin{bmatrix} C(u) \\ S(u) \end{bmatrix} \tag{4.1}$$

$$\begin{bmatrix} C(u) \\ S(u) \end{bmatrix} = \begin{bmatrix} \int_0^L cos(\frac{\pi}{2}u^2)du \\ \int_0^L sin(\frac{\pi}{2}u^2)du \end{bmatrix} \tag{4.2}$$

$$\begin{bmatrix} C(u) \\ S(u) \end{bmatrix} \approx \begin{bmatrix} 0.5 - R(u)sin(0.5\pi(B(u) - u^2)) \\ 0.5 - R(u)cos(0.5\pi(B(u) - u^2)) \end{bmatrix} \tag{4.3}$$

$$R(u) = \frac{0.506u + 1}{1.79u^2 + 2.054u + \sqrt{2}} \tag{4.4}$$

$$B(u) = \frac{1}{0.803u^3 + 1.886u^2 + 2.524u + 2} \tag{4.5}$$

In addition, in order to get a unique solution, appropriate restrictions must be imposed with respect to the desired outcome (Meek et al., 1989; Stoer, 1982; Meek, 1991). As no previous work on the extraction of the horizontal alignment of as-built sliding tracks exist, the subject is discussed in the context of road centreline alignment extraction that has gained quite a lot of attention lately due to applications such as road reconstruction and tunnel maintenance (Kang et al., 2014; Cheng et al., 2016), development of navigation and road guidance systems (Cremean et al., 2006), assessment of drivers' mental effort (Pellegrino, 2009), road safety and design compliance (Castro et al., 2005; Weigel et al., 2006; Zuriaga et al., 2010; Holgado-Barco et al., 2014), creation of road and railway inventory (Tong et al., 2010), advanced driver assistance systems (Jiménez et al., 2009), and virtual reality content creation (McCrae et al., 2009; Nguyen et al., 2014).

road alignment extraction

The traditional, manual process of designing a road alignment starts with an approximation comprising the tangents only, namely the control polygon; however, very few alignment extraction methods use it. The control polygon is a simple overall structure that is complimented by a bend at each vertex, so that the final alignment is of acceptable geometry according to road design

control polygon

guidelines and fits well in the physical environment. Walton et al. (2005) presented a way to calculate clothoids as blending curves given the vertices of a control polygon. The fact that these curves may lie further away from the polygon vertices was identified in Nguyen et al. (2014) as a drawback. This is beyond the point, especially in the context of their study, as it only implies that creating the curve is not as simple as fitting them to a subset of the input sample. On the other hand, the critique on the control polygon representation in Bauer et al. (2009, p. 726) is fair; indeed, extra care is required for bends at apex angles greater than 180° and overlap between adjacent segments must be explicitly avoided. Reliably identifying the sample points that belong to each tangent in order to construct the control polygon is also challenging. These are the main reasons why—together with the fact that this can become a large, non-convex optimisation problem—this approach is not popular even though it is arguably the most intuitive way for accomplishing this particular task manually.

The control polygon representation provides a simple and intuitive way to support global design guidance that allows local adjustments (Bauer et al., 2009, p. 725; Walton et al., 2005, p. 353). However, for most applications, simply representing the road alignment by a generic smooth parametric curve, i.e. a spline, which fits the data might be sufficient. For example, fitting a cubic smoothing spline (De Boor et al., 1978, pp.211–220) is a standard way to initialise a curve in a continuous form from an ordered list of discrete points; the user-defined smoothing weight may also filter the noise in the measurements. Another popular option is the locally weighted regression method (Cleveland et al., 1988) which may be used to remove a relatively small percentage of outliers from dense point sets by least-squares fitting of a first or second degree polynomial to the points in the vicinity of each point. For applications such as design compliance verification, simply estimating a few parameters for each road element independently, without actually creating a digital parametric representation of the entire alignment, is adequate. Nevertheless, in this case a good fit is not enough; the correct number and sequence of primitives need to be identified but in the presence of noise such estimations can easily be biased.

It can be argued that both approaches—selective parameter estimation and generic parametric representation—offer fast solutions to the problem of road centreline alignment extraction at the expense of reliability and accuracy. Another case where it is necessary to reconstruct the clothoid spline itself, rather than a simpler approximating curve, occurs during the design process of a roadwork or a track. The designer often has to create an initial alignment passing through defined landmarks that need to be traced manually, which is in fact a tedious and time-consuming process (Jong et al., 2003, p. 108; Lee et al., 2009, p. 217;Kang et al., 2012, p. 258). Even an initial ap-

proximation of the clothoid spline is more useful—especially if automated—as a base alignment, provided that it contains the appropriate number of geometric elements, compared to any generic smooth curve or simply a list—usually inconsistent—of some of its parameter values.

Extracting the road alignment from a discrete set of points, which was measured along its centreline by means of a surveying technique, is essentially a model selection problem. The concept of curvature-based geometry segmentation was briefly touched upon in section 3.5.2; it is now discussed in combination with curve fitting within the context of the horizontal alignment extraction process. In broad terms, there are two main tasks involved in most approaches that have been proposed so far. The sample is segmented according to a criterion related—directly or indirectly—to the curvature estimation of the underlying curve from a discrete point set; subsequently, a model of the appropriate type is fitted to each subset, followed by an optimisation step. This is quite a complicated (Garach et al., 2014, pp. 1–2), yet in this setting necessary, process (Gikas et al., 2012, p. 115).

4.2.1 Curvature Estimation

Many of the proposed road centreline extraction methods avoided solving a large optimisation problem. Instead, the parameters of each curve segment were extracted independently by manipulating its curvature plot with respect to the arc length, where slant lines correspond to clothoids and horizontal lines correspond to straight segments, if their curvature equals zero, or arcs, otherwise. An algorithmic framework for trajectory segmentation based on various spatio-temporal criteria was presented in Buchin et al. (2011). As the present work is not about vehicle trajectories per se, but rather on the geometry of the structures that accommodate it, the focus is on segmentation criteria related to curve curvature and shape fitting.

curvature

The curvature at any point along a smooth planar curve is a well defined property in differential geometry that measures how sharp the curve turns. More intuitive geometric definitions can be given based on the rate of change of the tangent direction, the reciprocal of the osculating circle radius (Calabi et al., 1998), or the deviation from the tangent line. Its invariance to rotation and translation has been exploited in many applications for shape matching, recognition, and registration (Mokhtarian et al., 1992). All the aforementioned definitions are equivalent in the continuous space, where curvature calculation is based on the derivatives of the curve's analytical or parametric representation. Extending this notion to the discrete setting is required in order to deal with datasets that do not have a simple analytical representation and are contaminated with noise, e.g. polygonal curves resulting from sampling continuous curves or scanning physical objects.

Differentiation is inherently and notoriously sensitive to noise (Holgado-Barco et al., 2014; Gikas et al., 2012; Flynn et al., 1989; Utcke, 2003; Liu et al., 2008; Lin et al., 2010; An et al., 2011) and a plethora of curvature estimation algorithms that attempt to overcome this drawback have been proposed (Lin et al., 2010; An et al., 2011; Taubin, 1991; Cazals et al., 2003; Belyaev, 2004; Lewiner et al., 2005). The question is whether or not they are all equally fit for purpose, which in this case is an estimation of the curvature sufficient to correctly initialise the segmentation of a sample collected along a road centreline to its constituent geometric primitive shapes. These methods can be broadly classified depending on whether they fit a continuous curve to the sample points in the vicinity of each point in a least-squares fashion, or attempt to directly approximate their differential properties from three or more discrete samples.

discrete

Most work on road alignment extraction did not go all the way in estimating the curvature. Instead, it stopped one step before that and only estimated the azimuth of its tangent at every point, using discrete numerical differentiation (Holgado-Barco et al., 2014; Tong et al., 2010; Gikas et al., 2012; Imran et al., 2006; Choi et al., 2007; Karamanou et al., 2009; Li et al., 2012). This approach either offsets the complexity to the segmentation part, as clothoid segments now correspond to parabolas, or compares deflection angles—differences between successive azimuth values—against a threshold which only works with points that are actually on or very close to the curve and are uniformly spaced. Otherwise, the distance between samples needs to be taken into account, which essentially is equivalent to calculating the second derivative, hence the curvature.

continuous

There is a variety of options when it comes to selecting a specific analytical or parametric model and the extents of a point's neighbourhood that will be used for the estimation of the curve's curvature at every polygon vertex. Generic parametric curves were usually preferred for the representation of a road's centreline due to their computational simplicity (Ben-Arieh et al., 2004; Romero-Zaliz et al., 2015). In this case, the segmentation step could be bypassed altogether by directly fitting one of the various types of spline curves, such as smoothing splines (Castro et al., 2006) or B-splines (Garach et al., 2014), to the full dataset, which should relatively dense in order to yield acceptable results. In essence, splines provide a way to convert a discrete set to a continuous representation and implicitly smooth it in a single step (Unser, 1999).

However, accurate curvature estimation from spline representations of road alignments can be problematic, as it can be affected by noise even for uniformly spaced point samples (Castro et al., 2006, p. 241) and lead to inaccurate primitive parameter value estimation. In addition, appropriate smoothing parameters, which are specific to each sample and do not generalise well (Garach et al., 2014; Castro et al., 2006), need to be defined for each type of spline. Even though transforming

the curvature estimation problem and solving it in the continuous space has practical advantages, it entails implicit global assumptions about the curve's curvature. Nevertheless, the curvature of such mathematical objects cannot be easily controlled (Nguyen et al., 2014, p. 262; Walton et al., 2005, p. 354; Havemann et al., 2013, p. 764) and the result can be greatly affected by the selected parameterisation scheme and anomalies such as outliers and significant gaps along the curve.

4.2.2 Alignment Segmentation and Extraction

Once the azimuth or curvature plot is constructed, the sequence of different primitive curve types—e.g. tangents, clothoids, circular arcs—along the centreline can be identified. The plot usually needs to be smoothed before proceeding to the segmentation task, either directly after creating it or indirectly by selecting the spacing between the points which will be used in the calculation. In Holgado-Barco et al. (2014), a moving least-squares filter was used, whereas Gikas et al. (2008, p. 401) used a median filter; its size had to be carefully selected so that it preserved the significant features. The smoothing parameter selection problem was addressed in Gikas et al. (2012), for the case of very small Gaussian noise, by using filter sets that are dynamically defined based on fractal theory. In Imran et al. (2006); Choi et al. (2007); Gikas et al. (2008); Tong et al. (2010); Li et al. (2012); Gikas et al. (2012), finite differences were used to calculate the tangents at each sample point combined with hard-coded, ad hoc angle thresholds for the segmentation. However, the sample-specific threshold values that were used can only be generalised for very accurate measurements and equidistant samples.

tangent identification

In Gikas et al. (2008), Holgado-Barco et al. (2014), and Garach et al. (2014), segments with curvature less than a pre-defined threshold were identified as tangents. This is a reasonable choice, but very sensitive to noise. It was assumed therein that the curvature plot corresponding to a straight segment lies entirely within this threshold; no sanity checks were performed, which means that either the process has to be supervised or the plot needs to be oversmoothed in order to avoid spikes being identified as bends. Nevertheless, oversmoothing will not only affect the absolute value of the curvature—and any indirect parameter estimations based on it—but also the boundaries of each segment. The values which were used as a lower bound to the radius of a segment in order for it to be classified as a tangent were 28648m, 11459m, and 1500–3500m, respectively. It was also assumed that these segments were in fact straight lines in object space and they were treated as such without further investigation. These assumptions were based on road/track design guidelines that propose such thresholds. However, these rules refer to dimensioning other design parameters, for example the crossfall, along an arc with low curvature and

the elements may very well actually be arcs. If they are long enough and their parameters are estimated locally, then ignoring their curvature can significantly affect the results.

Garach et al. (2014) fitted a cubic smoothing spline to points surveyed along a road centreline in order to estimate its curvature. They also proposed three pairs of values for the smoothing parameter and the truncation threshold based on the range of radii along a road segment which, however, are directly related to the noise and curvature of the specific data samples they studied. The parameter set of each curve segment was estimated in curvature space by area-based fitting of a trapezoid to each area between the curvature plot and two consecutive straight segments. The abscissae and ordinates of the trapezoid's parallel edges corresponded to the start and end stations for each element and the arc radii, respectively. The calculation was repeated after the transition lengths were corrected in order to take into account the bias due to truncation; S-bends were also considered but compound bends could not be detected. Only the length of each clothoid was calculated; if it was less than a given threshold the bend was considered a simple one and a rectangle was fitted instead of a trapezoid.

This two-step approach was also used in Gikas et al. (2012). The radius of each bend was initialised with the most probable value estimated from a curvature histogram, which was considered as more robust to noise compared to the maximum curvature. This value was optimised in a least-squares sense using the coordinates of the corresponding points on the arc. Subsequently, the refined radius estimation was used to indirectly estimate the clothoid parameter $A = \sqrt{RL}$, similarly to Imran et al. (2006). However, the initial clothoid length, estimated from the curvature plot, was used without correcting for the new radius. In a similar fashion, Holgado-Barco et al. (2014) used the curvature histogram and linear fitting in curvature space to segment the point set and initialise the parameter values. It has not been made entirely clear how exactly the least-squares fitting of the geometric elements was conducted but it was explicitly stated that least-squares adjustment of alignments parameterised using the radii of circular arcs has been unsuccessful (Holgado-Barco et al., 2014, pp. 222,226).

In the last three studies (Holgado-Barco et al., 2014; Garach et al., 2014; Gikas et al., 2012), the 2D coordinates of the points along the transitions were ignored and only the clothoid parameter A was estimated based on the radius change of the adjacent segments and its length, which was estimated in curvature space. This means that even a small error in the estimation of lengths and radii of the individual segments can have a detrimental effect if they were to be integrated back in object space (Baran et al., 2010, p. 656; Koegel et al., 2011, p. 90). This is exemplified in McCrae et al. (2009) where a dynamic programming algorithm was used to fit a piecewise linear approx-

imation to the curvature plot, estimate the clothoids' parameters, and align them in object space using rigid transformations. Even though Jiménez et al. (2009) went a step further in estimating the clothoid segments' object space coordinates, using common approximations for the Fresnel integrals, they did so for each bend individually without considering potential overlaps. Fixed tangents were used on either side in a constrained iterative scheme where all the parameters were estimated simultaneously. However, they essentially only optimised the circular arc's parameters using the points closest to its centre. Points along transitions were not taken into account for the model fitting, as the constraints that were used only provided for a unique solution for the clothoid parameters and guaranteed tangency; this could have easily been computed using a closed form solution (Kuhn, 2013).

arc splines

Conversely, Marinelli et al. (2017) claimed that the clothoid segments cannot be a priority in an alignment extraction process based on Bassani et al. (2016), who argued that estimating the clothoid parameters must begin with the parameter estimation of the adjacent arcs and tangents. The work of McCrae et al. (2009), Baran et al. (2010) and Nguyen et al. (2014), where each segment type, including clothoids, was fitted individually has showed that this is not the case. It is quite clear that ignoring the clothoid segments can be quite problematic if they actually exist in the centreline under consideration. Choosing to do so means that a different, well-behaved and potentially simpler, problem is addressed: that of arc spline approximation which has been formally studied by Meek et al. (1995), Bauer et al. (2009), and Maier (2014). Marinelli et al. (2017) were right in that line fitting is, in theory, computationally more stable, so it is a good choice to start by creating a control polygon; however, they did not consider near-parallel tangents whose intersection is extremely unreliable. In addition, if less than 3 points lied along tangents and less than 4 along arcs, the segment was simply ignored. Even though they identified the length of a segment as a critical parameter in model fitting, they presented results for tangents longer than 43.71 m and arcs longer than 177.35 m with 550 m radius.

In Tong et al. (2010), only a single bend sequence, tangent-spiral-arc-spiral-tangent, was investigated without considering a control polygon. The initial approximate values for the least-squares adjustment of the parameters were expected to be accurate and the segmentation was performed manually, just like in Karamanou et al. (2009); Gikas et al. (2008); Jiménez et al. (2009). In both Tong et al. (2010) and Marinelli et al. (2017), the potential misclassification of a sample point to the wrong primitive model type was handled by decreasing the observation's weight for the subsequent adjustment iterations. Without going into detail on the robustness of the scheme and the percentage of outliers it can handle, the main drawback is that miss-classified, yet accurately me-

asured, points were not taken into account. They also parameterised straight line segments using their slopes; an approach that does not lend itself to automation.

In Kang et al. (2014), instead of using the curvature plot in order to assign the appropriate model to point subsets for the extraction of a tunnel's centreline, linear, circular, and transition segments were directly parameterised using first, second, and third degree polynomials, respectively. Multiple parameter sets for each candidate model were initially generated. A model hypothesis for each one of the three types was estimated using a RANSAC scheme from the minimum number of points that was necessary for the estimation. The probability of the generated parameter set closest to the estimated parameter hypothesis at each iteration was increased and the one that reached a predefined threshold was used to initialise the corresponding model for each segment. Finally, a global adjustment was performed to achieve consistency between the overlapping segments, but this was based on constraints that do not guarantee G^1 continuity.

graph-based approaches

One of the most flexible approaches for clothoid spline fitting, targeted to the representation of sketched strokes, has been proposed in Baran et al. (2010). The curvature estimation was not directly used for primitive model type identification but rather in a curvature-based resampling scheme, which was applied to a hand-drawn polyline in an effort to reduce the sample size for performance enhancement. Curve segments with lower curvature were sparsely resampled and the initial polyline was segmented at points with large curvature. This is essentially a corner detection process that is suitable for the specific application of hand-drawn strokes where sharp corners are expected. The problem was formulated as a shortest path search in a weighted directed graph, where the nodes represented all the possible primitives types that were fitted to all combinations of vertices along each segment; the graph edges represented the different types of transition between each primitive pair. The selection of the primitive sequence was made through—rather unintuitive—weight assignments and a final global optimisation was performed to guarantee continuity between consecutive segments and proximity to the input. Each primitive was expressed in terms of its length, starting point, start-end direction, and start-end curvature, allowing clothoids to degenerate to arcs or lines.

Despite the large solution space, the graph-based implementation that was provided in Baran et al. (2010) is fast and suitable for real-time sketching. However, there are 9 user-defined and 23 internal parameters, plus 6 binary algorithmic options, that require tuning but—even though the authors claim that they all have geometric interpretation—selecting appropriate values is not straightforward and might never yield the desired result. An exhaustive graph-based approach was also used by Bauer et al. (2009) in fitting arc-line splines to represent the 3D axis of a bent

tube. Using a representation scheme that is common in bending machines, which is called Length-Rotation-Angle (LRA) and represents tangents by their length and rotation to the previous segment and the arcs by their apex angle and radius, they built a curve that comprises consecutive tangent-arc segments. However, in order to maintain local control, the candidate tangents were restricted to those between pairs of the sample points. As only a subset of the solution space is considered, even if a global optimisation process is used, the solution might still comprise more than the necessary segments (Bauer et al., 2009, pp. 723–724).

In graph-based approaches, curve segmentation and fitting is achieved by minimising the total number of primitives while maintaining a predefined maximum distance from the initial polygonal approximation and continuity between consecutive segments. This may cause over-fitting and yield over-segmented sequences—a characteristic of local fitting methods (Piegl et al., 1997, p. 363)—atypical for roadways, even if certain generic road design concepts are used to constrain the process, such as those introduced in Nguyen et al. (2014). They proposed a recursive algorithm that expands a curve along the initial polygonal approximation of a road centreline extracted from a geographic information system database—a greedy strategy previously outlined in Buchin et al. (2011). Starting from one end, a line, an arc, and a clothoid were fitted to all the remaining points and the one with the smallest residuals was selected. If at any point all options failed to conform to the predefined residual threshold, the remaining subset was split, based on a curvature-related metric, or even rolled back and started with a different primitive sequence. Extra points were added, if necessary, along the sparse polyline vertices at an interval calculated based on their spacing.

It is worth mentioning the approach which is used to deal with a similar problem from a macroscopic view for highway planning; for a comprehensive overview of the subject see Jha et al. (2006). Instead of seeking the best—in terms of geometric fidelity—curve that follows a discrete path, the aim is to discover the most economical path between two locations on a map, while adhering to a specific set of roadway design guidelines. A two-dimensional cost function is used to assign, to every location in the vicinity of the road, a value that is related to factors such as topography, land availability, environmental impact, earthworks and maintenance cost, and land-use (Jong et al., 2003; Kang et al., 2012; Jong et al., 1999). The main difference to the previously discussed methods is that the number of the control polygon vertices is used as a decision variable controlled by the designer (Kang et al., 2012, p. 268; Casal et al., 2017, p. 266) and that most initialisation heuristics are designed to avoid very costly solutions or restricted areas (Lee et al., 2009, p. 268; Casal et al., 2017, p. 266), instead of following a desirable path.

4.3 Surface Fitting

generalised cylinders

The most intuitive and widely used geometric representation of elongated objects is that of a generalised cylinder, which can be modelled using the spine- or path-based skinning method (Piegl et al., 1997, p. 462; Campbell et al., 2001, p. 173; Tang et al., 2010, p. 831). This is the process of blending together, usually planar, cross-sectional curves along a path, which serves as the spine, in order to create a smooth parametric surface in a continuous form. The path can be a space curve and the cross sections are, in general, perpendicular to it; for this particular application the cross sections are vertical and their horizontal projection is perpendicular to the path (Rempfler et al., 2016). This approach does not only simplify the design process but is also consistent with the along- and an across-track parameterisation that is required for the digital representation of the sled's trajectory on the track's surface within a virtual training system, q.v. section 2.2.2.

The path and cross sections are usually initialised in the simplest possible form and subsequently their parameters are optimised in order to fit a skinned surface that can be displayed and modified within a CAD software package (Bartoň et al., 2014; Yin et al., 2014; Dimitrov et al., 2016). Starting with a polygonal chain or mesh—which can be considered as the simplest continuous parametric form—the task of parameterisation assigns a coordinate system to the object that one wishes to describe and manipulate. It also provides an one-to-one correspondence between the surface and an image that can be used for texture mapping (Botsch et al., 2010). For the parameterisation of a point set that is already ordered along its generating curve, the most intuitive choice is its arc length. As this is in general unknown, several approximations have been proposed such as the uniform, cumulative chord length or the centripetal parameterisation method (Hoschek, 1988; Piegl et al., 1997, pp. 364–365). For relatively flat surfaces, the parameters of each point can be initialised by projecting them on a plane. However, for 3D objects that cannot be represented as a function by a 2.5D surface, using a simple plane for the initialisation of an optimisation scheme is not ideal. Subsequently, the fitting problem is reduced from 3D space to a two- and one-parameter domain for surfaces and curves, respectively (Piegl et al., 1997; Barentzen et al., 2012).

parameterisation

Skinning—also known as lofting if no spine path is used—has been studied since the seventies within the shipbuilding, automotive, and aircraft industries and during the eighties and nineties within CAD software using B-splines for surface interpolation (Filip et al., 1989; Piegl et al., 1997, pp. 457–458). In the simplest case of a single cross sectional curve, the process is reduced to a translational sweep. The position and orientation of instances of the cross-sectional curve—which are isoparametric in the resulting surface—at intervals along the longitudinal direction is defined

skinning

by the path's geometry. The control points of the surface are generated by interpolation between the cross sections' control points. Even though conceptually simple, a robust skinning implementation is quite difficult. Using existing commercially available software is also quite restrictive as there is limited control over the interpolation between the section curves along the spine (Piegl et al., 1997, pp. 461–462). Several problems that were related to minimising the number of control points, making the knot vector homogeneous (Wang et al., 2008), cross section blending (Choi et al., 1990), and twisting and scaling (Marhl et al., 1996) have been overcome over the years but more came up when approximation algorithms were required in order to deal with datasets from scanning devices for reverse engineering applications (Ueng et al., 1998). Initially, the aim was to get a smooth surface that could approximate the input within a predefined tolerance (Piegl et al., 1996). Nevertheless, traditional approaches cannot effectively handle badly parameterised cross sections or rational curves, in which case more control points are necessary (Piegl et al., 2002).

One of the main problems for both interpolating and approximating algorithms is filling in missing information. In Tokuyama (2000), for example, missing cross sections were compensated for by constraining the longitudinal curves with respect to the geometry of the path. In general, small holes in meshes are treated locally using context based methods (Harary et al., 2014). Recently, path-based skinned surface fitting methods re-attracted attention as a means of reconstructing models from incomplete points clouds, q.v. figure 4.5. Wu et al. (2012) presented a framework for the schematic representation of buildings using image-based sparse point clouds as input. The horizontal boundary edges of each floor plan, called transport curves, were used as the skinning paths. The vertical slices, representing the walls, were expected to be simple and relatively straight. Initially it was assumed that all the slices shared a common profile; by accumulating all the slices, including the incomplete ones, a cross section could be created. For transport curves with more than one cross sections, the initial cross section was used as a seed in a clustering method in order to recover more section that were significantly different. However, no smooth transitions were considered along the path, which resulted in sharp corners representing walls parallel to the slices. Finally, a global optimisation process was performed in order to blend together the network of transport curves and cross sections in order to create a smooth detailed surface of the building.

In contrast to Wu et al. (2012), where the object of interest could certainly be represented by a sweep surface, Bartoň et al. (2014) attempted to detect whether parts of any given free-form object could be approximated by sweeping, q.v. figure 4.6. However, their exhaustive search was time-consuming and could yield numerous overlapping patches created from open-only cross section curves along different paths. Conversely, Yin et al. (2014) presented a curve-based interactive

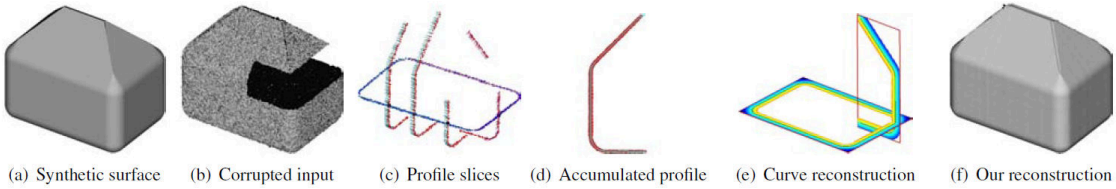


Figure 4.5: Schematic surface reconstruction; reproduced from Wu et al. (2012, fig. 4).

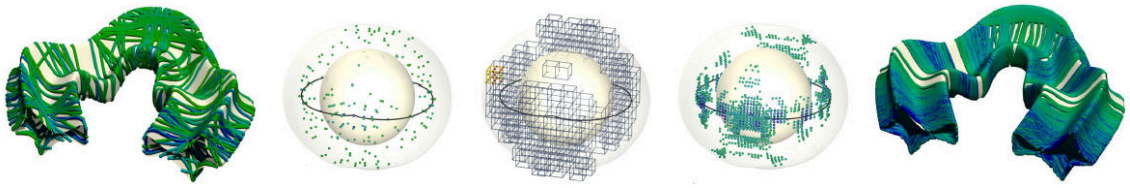


Figure 4.6: Detection and reconstruction of free-form sweeps; reproduced from Bartoň et al. (2014, fig. 7).

method for the reconstruction of small objects, which were scanned in their entirety as watertight surfaces, q.v. figure 4.7; the point clouds were expected to have gaps due to occlusions but no extraneous points or outliers. The process was based on extracting their skeleton and using it as the skinning path, which should be manually refined. Point slices were extracted along each skeleton branch but only a few of the closed curves were used as cross sections according to a completeness measure that is based on the angular gap between successive projection of the point cloud vertices on the extracted cross-sectional curves. Morphing between cross sections was performed in an as-rigid-as-possible fashion while minimising the sum of the distances from each points to the closest cross-sectional curve. A global interpolation scheme was finally used in order to create a smooth sweep between successive cross sections along each skeleton branch which had to be interactively enhanced.

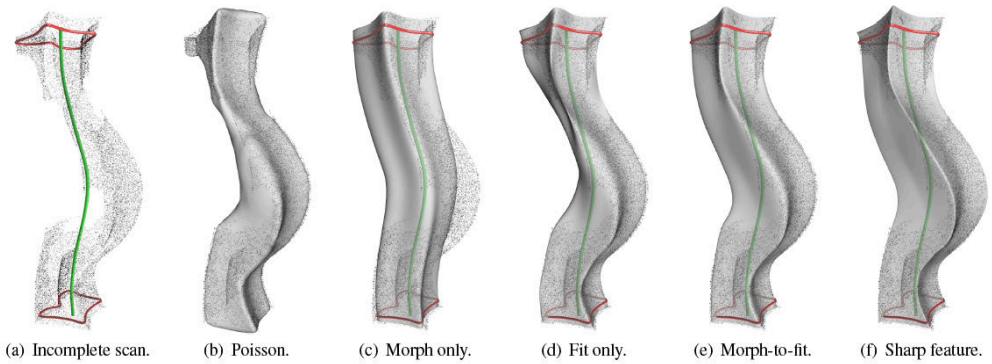


Figure 4.7: Morfit: interactive surface reconstruction; reproduced from Yin et al. (2014, fig. 2).

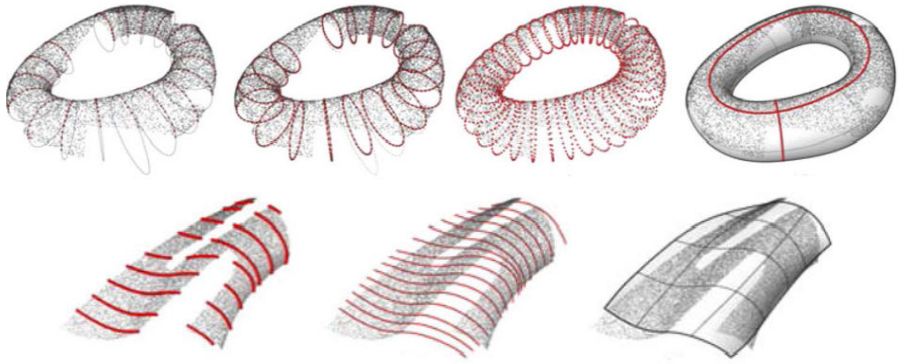


Figure 4.8: NURBS fitting for as-built modelling; adapted from Dimitrov et al. (2016, fig. 13, 15).

Dimitrov et al. (2016) proposed a method that is targeted at as-built modelling of pipes and buildings using B-splines, q.v. figure 4.8. Noisy point clouds need to be segmented to a certain level which is defined by a user-defined parameter; given a subset that corresponds to a single surface, cross-sectional planes were randomly generated and the best one was selected based on whether the estimated point normals were parallel to it or not. Open curves were considered to be partial and were extended according to their curvature near the endpoints, which means that no attempt to detect the surface boundaries is made. Finally, a non-uniform rational B-spline (Piegl et al., 1997) surface was created by lofting over the cross sections without using an alignment, which means that large gaps, perpendicular to the cross-sectional planes along curved parts of the object, may affect the surface geometry.

Chapter Summary

In this chapter, the importance of the selection of an appropriate geometric representation scheme was discussed. Representing a track as a generalised cylinder is not only well-aligned with the design intent of such objects but also simplifies the transformation to other forms as required. The most important component of this representation for a track is a clothoid spline representing its centreline alignment; however, existing approaches that automate the extraction of a clothoid spline from a planar point clouds were not designed for point sets with significant gaps and noise. In addition, surface fitting methods that rely on similar representation schemes either require user interaction or are tailored towards closed and watertight surfaces and implicitly fill any gaps instead of detecting potential surfaces boundaries, therefore cannot be used for track reconstruction without significant modifications.

Chapter 5

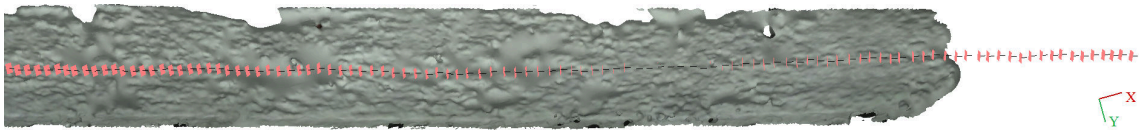
Preliminary Investigations

In the first instance, a preliminary investigation was necessary in order to assess the reliability and completeness of the results that could be obtained from datasets that were collected during track walks, without rigorous photogrammetric planning, using off-the-shelf equipment. These datasets could provide an insight on the complexities involved in the process of developing an automated and targetless photogrammetric workflow for this specific scenario. Video streams of a track walk on the Whistler track and images acquired along the Winterberg track were provided for this purpose. In this first attempt to reconstruct a track using image-based scanning, all the datasets were processed in PhotoScan (2018), q.v. section 3.6.1.1. No other measurements that could unambiguously define the scale and orientation of the track model were made at the time of the data acquisition. In addition, no details were available about the internal geometry of the consumer grade cameras' that were used, as they were not calibrated.

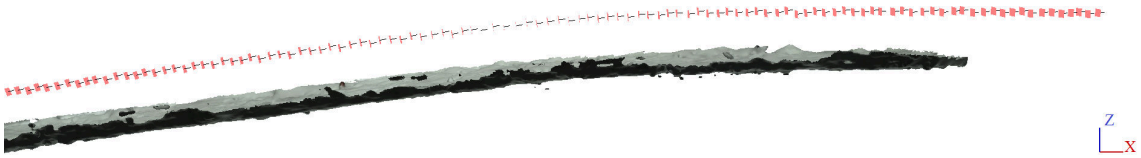
5.1 Whistler Track Reconstruction from a Track Walk Video

The images that were used for the reconstruction of the Whistler track originated from a 19 minute long video file with a resolution of 720×576 pixels, which was provided by the University of Nottingham. It was acquired during a track walk on ice using a ©SONY DCR-SR300 (2017) camcorder with a 4.96×3.72 mm sensor that recorded at a rate of 25 frames per second (fps) and was equipped with a progressive shutter and a zoom lens with focal length that could vary between 5.4 and 54 mm. During the video recording, the camcorder was moving in a linear fashion along the track while maintaining an approximately constant height from the floor and an approximately constant roll angle, q.v. figure 5.1. Its path was almost parallel and equidistant to both walls at the straight segments of the track and close to the inner wall at the bends, where the flat part of the

dataset



(a) Top view of the starting straight.



(b) Side view of the starting straight.

Figure 5.1: Camcorder trajectory and video frames extracted at 5 fps along the starting straight of the Whistler track.

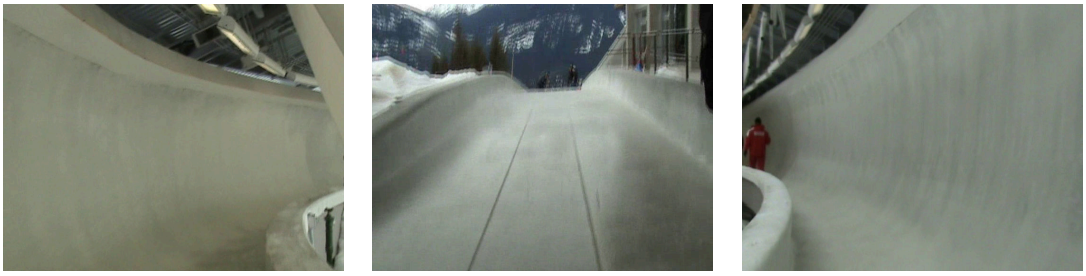


Figure 5.2: Blocked views of the Whistler track.

track is accessible on foot.

Two datasets, comprising 2288 and 5680 images in the portable network graphics (PNG) file format, were extracted from the windows media video file at 2 and 5 fps, which corresponded to a recording interval along the track of approximately 0.6 and 0.25 m, respectively. Almost 20% of the images in both sets included people moving or standing at certain parts of the track—either maintenance workers or athletes performing track walks, q.v. figure 5.2—or were significantly blurred due to the cameras' motion or being out of focus. The quality of each image was estimated using PhotoScan and about 7% of them with a quality metric value below 0.5 were removed, after visually confirming that they were indeed blurry. Considering the overall image quality, the recommended cut-off value was not increased as it might lead to removing perfectly good frames and the image alignment to fail. Figure 5.3a, for example, is obviously blurrier than figures 5.3b and 5.3c but was assigned a higher quality metric. An additional 3% of the images were removed after visual inspection, as the view of the track was fully blocked by one or more people standing in front of the camera or they were obviously blurry, even though they were assigned a quality metric value above the recommended threshold.

The camera's principal distance was initialised to 7 mm for the image matching process which was performed at full and a quarter of the resolution for both datasets, without setting an upper limit on the number of the detected and matched feature points, and lasted 30 minutes and 2.5



(a) 0.59 **(b)** 0.68 **(c)** 0.51

Figure 5.3: Image quality estimation for three blurry video frames of the Whistler track.

hours, respectively; the image orientation stage that followed required approximately the same amount of time. On average, 11850 points were detected on each image and fewer than 1000 were matched on each pair. The result, for each dataset, comprised 3 separate blocks of relatively oriented images, as no matches were found between successive images at two of the places along the track where the camera was moving through groups of people, q.v. figure 5.2. Several attempts to facilitate the automatic image matching process by using better guesses about the camera's interior orientation were also unsuccessful. In a last attempt to recover the track's geometry, the entire dataset for each frame rate was processed, including all the extracted images except for those with a quality value of less than 0.3; as a result, the relative orientation of the entire set that was extracted at 5 fps was successfully recovered. Apparently, the image redundancy was able to overcome the effects from the image blurring and the partially obstructed field of view.

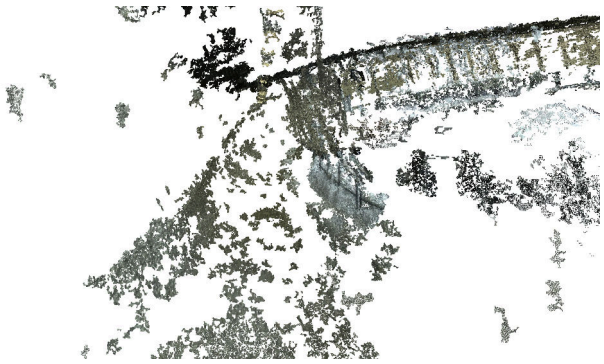
Subsequently, a free self-calibrating bundle adjustment was performed, using image data only as no ground control points were available, in an iterative fashion in order to remove outliers—i.e. feature points with reprojection error greater than 1 pixel, projection accuracy greater than 25, and reconstruction uncertainty greater than 1000, q.v. section 3.6.1.1. Finally, object-space scale information based on the track's nominal width along the straight segments, q.v. figure 5.4b, were used in a bundle adjustment in order to fix the scale along several parts of the track. A dense point cloud of the scene was extracted using the images at full resolution. As the frames were blurry and of low resolution, an aggressive filtering option was used in order to remove noise and outliers and only retain reliable points. The dense cloud depicted in figure 5.4a is obviously of very low quality and density. Gaps were partly compensated for during the mesh generation process; however, geometric features that did not represent the track's surface were also generated due to noisy depth estimations, q.v. figure 5.4d.



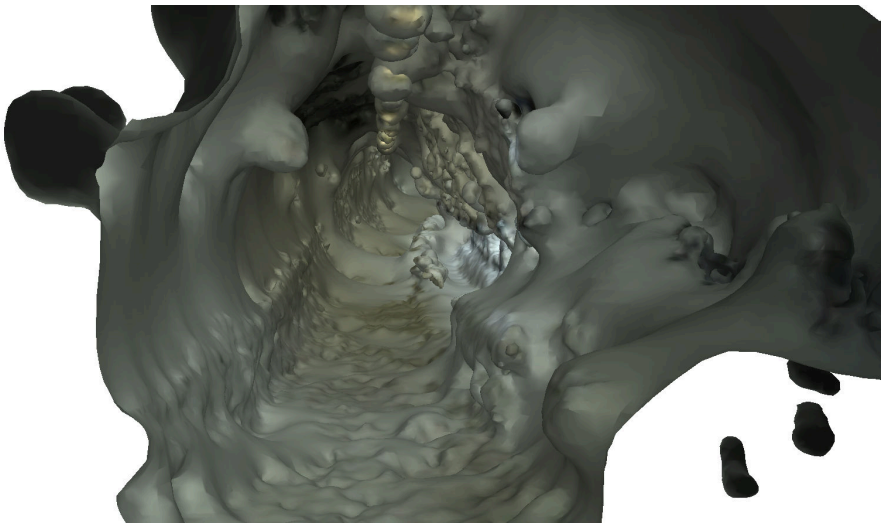
(a) A perspective view of the reconstructed dense point cloud.



(b) Two of the manual observations that were used in order to define the photogrammetric model's scale by constraining wall-to-wall distance at the straight segments to the track's nominal width.



(c) The dense point cloud from the point of view of the image in figure 5.4b.



(d) The mesh that was generated from the dense point cloud in figure 5.4c.

Figure 5.4: Dense reconstruction of the Whistler track.

5.2 Winterberg Track Reconstruction from Track Walk Images

An off-the-shelf, uncalibrated ©Panasonic Lumix DMC-G1 (2017) camera with a 14 mm minimum nominal focal length and sensor size equal to 17.3 × 13 mm, was used to acquire 650 images with 2816 × 2112 pixels resolution along the Winterberg ice track. The image sequence was provided by the English Institute of Sport (EIS) and captured by a person without previous experience in photogrammetry, following a set of simple instructions for the acquisition of stereo pairs at 2–4 m intervals along the track.[1] The data collection lasted a little over 1 hour in total but was interrupted halfway through due to scheduled sled test-runs. The principal distance for the image matching process was initialised to 14 mm according to the exchangeable image file format (EXIF) information. The same processing workflow as before was followed, q.v. section 5.1, however, the relative orientation of the complete image set as a single camera network was not possible. Instead, 3 disjoint blocks were retrieved comprising 257, 246, and 147 images, q.v. figure 5.6. The estimated along-track intervals between successive stereo pairs and the stereo base for each pair in the first block are shown in figure 5.5; on average, the former were 4.5 m and the latter were equal to 0.65 m. The dense point cloud along the first block was generated within 6 hours using the images at full resolution but parts of the track floor and the walls along several segments of the track—including along the starting straight, q.v. figure 5.7—could not be fully reconstructed.

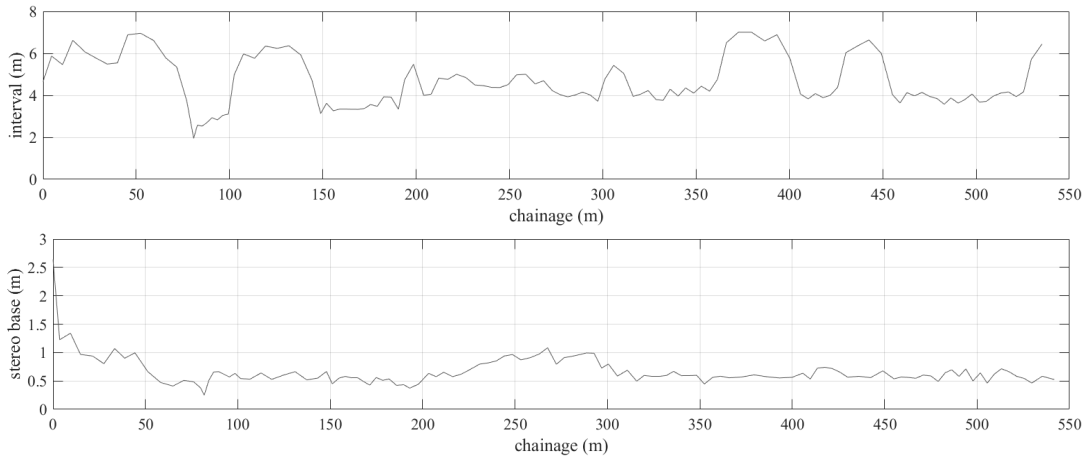


Figure 5.5: Along- and across-track distances between stereo pairs with respect to their position along the first 550 m of the Winterberg track.

[1] In the context of this thesis, where a hand-held camera is considered, the term stereo pair refers loosely to two images with almost parallel principal rays and a stereo base nearly perpendicular to them; not to be confused with strict stereo-normal image pairs.

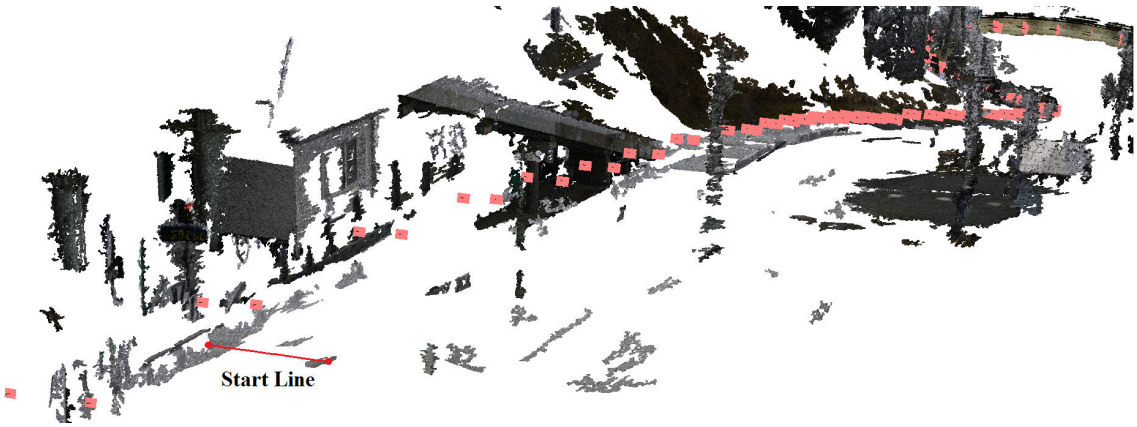


(a) Start **(b)** Middle **(c)** Finish

Figure 5.6: Sparse point clouds of three disjoint camera networks of the Winterberg track, cf. figure 2.2.



(a) A stereo pair at the startintg straight—the one at the lower left corner of figure 5.7b. (○) detected features, (●) matched features.



(b) The reconstructed dense point cloud of the startintg straight.

Figure 5.7: Perspective view of the dense point cloud and stereo pairs along the starting straight of the Winterberg track.

Chapter Summary

The technology that is required to digitally reconstruct a real object is available to any interested user, either as a commercial product or as an open-source solution. Nevertheless, each object-sensor-platform combination presents different challenges and may require a certain technical skill set from the operator for data collection and processing. Geometry processing methods are categorised according to the type of problem they solve; they are specific to the surveying method that was used for the data acquisition and the geometry of the object of interest (Berger et al., 2016, p. 2), which means that none of the existing ones may be readily suitable for creating the 3D model of a sliding track in the setting that was described in sections 1.1 and 1.2.

Some of the challenges, which are primarily related to the shape and the reflectance properties of the track, were demonstrated in the two track scanning tests. Two standard imaging configurations were used for image acquisition with off-the-self uncalibrated cameras. Even though parts of the scene were reconstructed, generating the complete surface of the entire track was not possible. In addition, the parametric 3D modelling of the parts that were reconstructed, would require significant manual effort in order to compensate for the missing information and remove noise and outliers. Considering the lack of prior knowledge on the subject, any surveyor would require additional information and reference measurements; however, in this scenario, such information may not be available or attainable. In summary, it was indicated that the automated image-based reconstruction of a 3D mesh representing an ice track and its surrounding environment is possible but not every attempt may be successful. In any case, any 3D modelling workflow would require significant manual effort in order to compensate for irregularities in the input mesh or point cloud.

Chapter 6

Synopsis

The review of the relevant literature allowed for the aim of this study, q.v. section 1.2, towards exploring the potential of designing a robust, automatic workflow for the image-based extraction of the parametric 3D model of a track—which can be carried out by non-experts using low-cost, off-the-shelf, hand-held equipment—to be broken down into specific objectives, which can lead to contributions of scientific and practical significance within a clearly defined domain. Fulfilling this aim will be achieved by reaching the objectives that were defined in section 1.3; a summary of the issues they address and an overview of the corresponding methodological approaches is outlined below for each one.

With the increasing competition in sliding sports, it is only a matter of time before track scanning using hand-held sensors becomes an integral part of every track walk. Ideally, an athlete or manager should be able to generate a fit-for-purpose 3D model of the track using an automated process. The question is how and under what circumstances such a model can be generated and whether there exist simple metrics that can convey its quality, in the case where no rigorous assessment is possible, so that its fitness can be assessed for further use within a virtual reality system. As the performance of any scanning method is affected not only by factors related to its own intricacies but also by the object itself—in particular its size, geometry, and surface properties (McGlone, 2013, p. 1049)—any indication of the empirical accuracy of image-based scanning of objects of different geometry and appearance may be misleading. Despite the elaborate literature review, reports on the performance of close-range photogrammetric techniques on elongated objects made of ice are rare, q.v. section 3.2.3. An indication of the achievable track reconstruction accuracy using traditional surveying techniques was given in section 2.3, based on the Whistler track construction study (SAIT, 2012). Significant time and effort, expensive equipment, and a surveying team with high level of expertise were required on site and in the office, in order to

achieve a reconstruction accuracy no better than 10 cm.

Methodological approach towards objective №1 - Track-scanning benchmarks: There exists a huge number of combinations of feature detectors and descriptors, matching strategies, and dense image matching algorithms—multiplied by several implementations for each one—to choose from for image-based 3D reconstruction but no conclusive studies on the superiority of any one of them over the others, q.v. sections 3.3.2 and 3.3.3. Comparing all the available imaging equipment and software solutions is practically infeasible—and beyond the scope of this work—especially as the related technology is improving at a fast pace; any conclusions from the comparison of a particular subset of them would be trivial. In order to provide a stable basis towards developing a proof of concept for the proposed techniques and workflow, a DSLR camera—real or virtual—and PhotoScan (2018) will be used as an "instrumental combination of camera and image measuring device" (Luhmann et al., 2013, p. 357) for the experiments which are presented in this thesis. A track site will be surveyed using both laser and image scanning techniques. Additional reference measurements, with better accuracy by at least one order of magnitude, in the form of ground control points will be collected on-site. The along-track geometry of the results will also be compared to alternative methods that do not require access to the track site.

Currently—and in the general case for the foreseeable future—project planning and as-built 3D modelling are tedious tasks that benefit from automatic procedures but require human interaction throughout. Both tasks are equally important, as the completeness of the reconstruction and the quality of a parametric 3D model depends not only on the quality of the input data, e.g. the images, but also on the strategies which are used during data acquisition and processing (Hoppe et al., 2012, p. 2). The datum definition is the first stage of the network design process but the configuration problem which follows is the most complicated one, q.v. section 3.4. The second objective considers the combination of two topics which can afford further improvement according to Schlogelhofer (1989), Barazzetti (2011, p. 131), Fathi et al. (2015, p. 157), and Rumpler et al. (2017, p. 256): explore best practices for image-acquisition guidance with respect to standard geometries. Imaging configurations that involve a forward-facing camera are avoided when dense reconstruction is the primary aim, thus not well studied. In any case, for applications where at least one forward-facing camera is used, such as visual SLAM or visual odometry, the trajectory of the moving platform that carries the sensors is by no means restricted in real time with respect to an optimal imaging configuration for dense reconstruction.

The main problems a designer faces in performing the simulations are related to the facts that network design relies on hard-to-conceptualise human expertise, the selection of which object fe-

atures will be observed on an image is subjective, and that realism is hard to achieve—especially with respect to the imaging process (Mason, 1994, pp. 21–23; Mason, 1995, pp. 14–15). Not only is decoupling the interior and exterior orientation effects on self calibration not always possible, q.v. section 3.3.1, but the realistic simulation of image distortions is challenging, as well. CAD, 3D modelling, and rendering software can assist with the realism, especially in close-range applications but—if, for example, they were to be used in order to extend the conclusions from the well-studied aerial network designs to UAV imagery—the texture map that is applied to the virtual 3D model of a scene should at least have a representative ground sampling distance, which may be hard to generate for large image scales. What Slocum et al. (2017) essentially suggested in their simulation example was replicable results on unrepeatable scenes.

Alsadik et al. (2013) have correctly identified the need to be able to accurately implement the network design on site; however, a workflow that requires the user to replicate rendered views on site can be quite cumbersome and time consuming. Their simulations were based on the assumptions that a coarse 3D model from features points which were matched along a video sequence of the scene can adequately represent the object's geometry. They also assumed that a static representation of the scene's geometry is adequate; as the camera's optical axis in all views is considered to be approximately perpendicular to the observed surfaces, all points should be observable unless self-occlusions are detected. However, the features that will be detected on high resolution images during the actual survey may have a vastly different density and distribution, especially in non-signalised scenes with glossy surfaces and small incidence angles.

Methodological approach towards objective №2 - Photogrammetric network design: Even though the design by simulation strategy may not lead to the best possible result, due to its trial-and-error nature, its effectiveness has been proved in practice and, in any case, "it is foolhardy to undertake any essentially new project of photogrammetric triangulation without first subjecting it to a thorough process of computer simulation" (Brown (1980) in Mason (1994, p. 21)). The intent, with respect to network design for track scanning, is not to shift the responsibility to the user by creating yet another tool that could potentially yield an imaging configuration that cannot possibly be implemented on site but to find the best amongst several simple configurations for a forward-facing camera; that is, an imaging configuration which may not be optimal in a formal photogrammetric sense but can achieve full track coverage at an acceptable quality within the allocated time. In order to do so, computer simulations will be performed using computer generated and real objects which have the geometry of a typical track. Image sequences, collected on site and rendered from a 3D model, will be processed within a common image-based scanning workflow

in order to identify factors that affect the quality of the reconstruction and to limit the search space for the on-site track scanning trials.

Even for an overdesigned photogrammetric survey, there can be no guarantees for a perfect result. In this case in particular, where specular and translucent surfaces with poor texture may be involved, a dense point cloud with significant noise, gaps, and outliers is to be expected as a result of the dense image matching or laser scanning process. Unnecessary information will almost certainly be included while parts of the track's surface will be missing; therefore, additional effort will be required for the tedious manual 3D modelling process in order to convert an initial implicit geometric representation to a form suitable for the representation of a virtual environment. With regard to the track itself, a suitable form refers to a parametric surface of the complete extent of the track that is covered with ice, with or without the sidewalls. Identifying the boundaries of the track is essential not only for the application of the appropriate material and texture to its 3D model. An important implication arises from the fact that the estimation of the sled's trajectory on the track is based on interpolation between a predefined number of surface patches, q.v. section 2.2.2; if the lengths of the cross sections significantly differ, striking a balance between performance and oversmoothing becomes challenging. The track segmentation can be achieved in 3D space using geometric cues such as point density and curve or surface smoothness and curvature, q.v. section 3.5. This is a well-studied approach which, however, requires a user-defined parameter in order to guide the process based on implicit assumptions about the nature of the point cloud.

Methodological approach towards objective №3 - Track surface modelling: More than one options exist with respect to the 3D modelling workflow that is most appropriate for elongated objects, q.v. figure 4.1. Even though sensor-specific information which may be beneficial to the process will also be examined, a general treatment of the track reconstruction process will be pursued. The generalised cylinder representation has been successfully used as the basis for the reconstruction of objects of similar geometry, q.v. section 4.1; therefore, it is expected to perform well if appropriate curve models, which can adequately represent the track's alignment and cross-sectional curves, are extracted from a 3D point cloud or mesh. In addition to removing extraneous parts of the acquired geometry, the track's model needs also be enhanced by attributes that are related to the simulation process. To this end, the surface extraction process will be complemented by an automatic start line detection process. The process will also be extended in order to generate a track model with the correct size and orientation, with respect to the earth's gravity vector, from photogrammetric models with arbitrarily defined datum.

Fitting an appropriate curve model along the track, in order to represent its alignment, can com-

pensate for missing across-track information by sweeping typical cross sections along its length. The most appropriate curve model for the track's horizontal alignment is that of a clothoid spline. The performance of existing techniques for horizontal alignment extraction was demonstrated on point sets with almost uniform and dense sampling rates, contaminated with small Gaussian noise and no missing data, which is rarely the case when it comes to real-world applications, q.v. section 4.2. Commercially available road design software have integrated automated best-fit alignment facilities for almost a decade now (AutoCAD Civil 3D, 2018; MXROAD, 2018). Nevertheless, simple trials have shown that they are effective only with well-behaved and appropriately spaced data with low levels of noise, as their main premise is that road design is a manual process that relies on human interaction. For example, ©Autodesk's AutoCAD Civil 3D (2018), one of the most popular commercial software for infrastructure design, offers a function that attempts to automatically fit an alignment to a curve or point set; however, it will not even process the former if it contains an intersection, i.e. a central angle greater than 180°, and suppress it for the latter.

Methodological approach towards objective №4 - Horizontal track alignment extraction: An algorithm that replicates the manual design process will be developed, so that cause and effect can be easily identified in any failed attempt, but without the need for user intervention. The familiar curvature plot and control polygon structures will be utilised so that the result is interpretable and, if necessary, allow for further adjustments to the result. Its performance will be tested on computer generated data along curves with challenging combinations of road elements and on data extracted from track scanning trials.

Track scanning involves the generation of a potentially large point set, whose density is defined by the requirement for the track surface resolution, which is higher than that of the surrounding environment. Eventually the part of the 3D point cloud or mesh that corresponds to the surrounding environment will be segmented and its density reduced to a more manageable level appropriate for real-time graphics generation. For the image-based scanning process, in particular, in order to avoid generating the point cloud of the entire scene at a high density, and potentially speed up the dense matching process, masking the images may be a solution.

Methodological approach towards objective №5 - Track image segmentation: A common workflow for the segmentation of the track's concrete structure whether it is covered with ice or not, is desirable; therefore, material recognition methods will not be considered. On the one hand, because the signature of a glossy ice surface would be tightly connected to its surrounding environment, thus, hard to generalise not only to a different track but to a different segment of the same track, q.v. figure 1.2b. On the other hand, a frosty ice surface would be hard to differentiate

from the snow on the surrounding environment even if they appeared as two disconnected areas on an image. Therefore, the proposed method will rely on the full automation of a state-of-the-art, interactive graph-based technique for foreground-background segmentation, q.v. section 3.5.1.

A practical problem with offline photogrammetric or laser-scanning techniques is that the integrity of the data can only be verified, if at all, during the processing step, after the acquisition is finished. It has been established that the limitations with respect to dense surface reconstruction which are imposed by real-time visual processes are too restrictive, e.g. low resolution; however, a completely offline measurement mode is not ideal either, q.v. section 3.2.1. Even if data verification can be performed on site, right after acquisition, repeating an imprecise measurement is not practical; it may not even be possible in cases such as this where the scene is susceptible to dynamic changes and access may only be granted for a limited amount of time. The verification process cannot be too strict, either, under such unfavourable conditions, as time constraints only allow for the collection of the least possible amount of data that will be necessary for the reconstruction. Anticipating and dealing with occlusion and moving objects may be quite straightforward for a human operator but identifying a blurry image is quite hard on a small low-resolution screen of a DSLR camera.

Methodological approach towards objective №6 - Online feedback on image quality: It was assumed that the main reason why the preliminary trials, q.v. chapter 5, failed to yield a complete track reconstruction was the nature of the image sets; in particular, the relatively low resolution, poor quality, and unstable camera geometry. An online feedback loop can potentially provide valuable information about the quality of the images during acquisition. In order to fulfil objective №1, high resolution images will be acquired with precalibrated cameras; as a stable sensor geometry requires camera functionality such as image stabilisation to be disabled and handheld photography in combination with higher resolutions and poor lighting conditions may call for longer exposure times, images of low quality are to be expected. Given that high image resolution and overlap alone, cannot guarantee that enough features matches for the relative image orientation will be generated, potential problems in these sequences will be studied with respect to the quality of the images, to the extent that it can be conveyed by metrics that have been proposed in the literature.

Part III

Episteme

This part presents the methods that were designed in order to fulfil the aim and objectives of this project, as well as simple experiments that demonstrate their functionality—two large scale trials are presented in part IV. Chapter 7 discusses the photogrammetric network design process for track scanning; the imaging configuration problem is addressed, in order to guide the image acquisition process with a hand-held camera. In addition, two methods are presented for the extraction of information from the collected images that will facilitate the subsequent 3D modelling phase. Chapter 8 is concerned with algorithms that will automate the 3D reconstruction and modelling process. A method for the extraction of the track's surface from a 3D mesh with noise, clutter, and missing parts, in order to feed a lofting or path-based skinned surface approximation process, is presented in section 8.3. It is based on a clothoid spline fitting algorithm, q.v. section 8.2, for the extraction of the track's horizontal centreline alignment. The datum definition problem in the absence of reference information is also discussed in section 8.1.

Chapter 7

Image-Based 3D Scanning

Image-based 3D scanning describes the process of generating a 3D point cloud from images of a scene based on its reflective properties and the geometric relationship between the images and points of interest. This chapter covers photogrammetric computer vision elements that will facilitate the image acquisition process and the subsequent knowledge intensive task of geometry processing, q.v. chapter 8. More specifically, the photogrammetric network design problem is addressed in section 7.1. The reconstruction capabilities of simple imaging configurations, which can easily be realised on-site with a hand-held camera, are assessed based on computer generated and real objects. The automatic segmentation of the track and the detection of its start line on images which were collected along its length using the aforementioned imaging configurations, are discussed in sections 7.2 and 7.3, respectively.

7.1 Network Design Simulations

One of the first questions that should be answered before embarking in a photogrammetric survey is "how should one collect the images?". Hence, the topic of this section is first order network design, q.v. section 3.4. In the general case, the correct answer to this design problem is to create a convergent imaging configuration with the viewing direction at each camera station approximately perpendicular to the surface that is to be measured. In order to illustrate what the aforementioned notion entails for an elongated object, two simplified networks based on the stereo normal case for the image-based scanning of a flat area are depicted in figure 7.1. A full frame digital camera with a 14 mm lens is considered, as it is wide enough for close-range applications but without extreme distortions along the edges of the image frame. Assuming that the camera is held by a person 1.5 m above the surface and a moderate overlap of 60%, 850-1250 images at 1–1.5 m intervals would

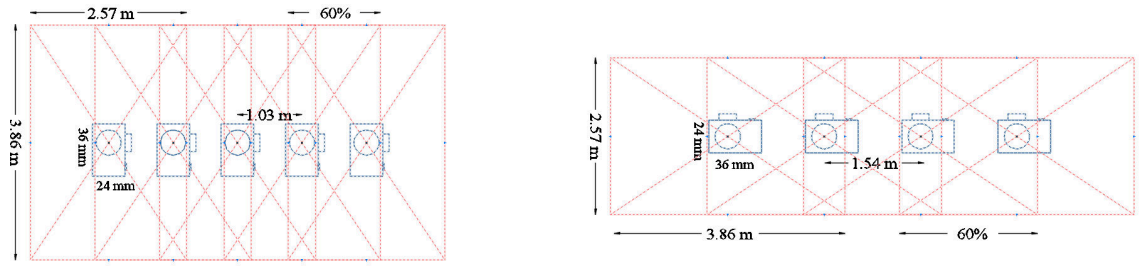


Figure 7.1: An illustration of the number of images that are required in order to cover a flat surface at a distance of 1.5 m from the cameras with two stereo-normal image blocks with 60% overlap.

be required for an area approximately the size of the Winterberg track, that is 2–3m×1300 m. Note that achieving full coverage of the surrounding environment, which is essential in creating a complete virtual environment, and around the corners—where the track surface lies on either side of the camera's trajectory—additional images are required. A convergent network would require even more images but that is already 30–100% more than the—presumably, too few—images that were captured during the preliminary trial in Winterberg, q.v. section 5.2.

When space and time are limited, the image number can be reduced if oblique views of the track are acquired, with the limiting case being that of a forward-facing camera having its the principal ray approximately parallel to the track floor. It is clear that several problems, related to the varying ground sampling distance and projective distortions, arise from such configurations with respect to image matching, point triangulation, and dense reconstruction. As there is an unlimited number of possible imaging configurations, a set of typical practical cases is considered, in order to generate general rules for economical acquisition of high-resolution images in confined spaces. The hypothesis was that there exists one among several imaging configurations of canonical geometry, which are simple enough to be implemented on site with a hand-held camera, that can yield a reconstruction significantly better than the rest, within the spatial and temporal constraints of a track walk. In order to investigate the existence of such a camera network, three experiments were designed, which gradually increase the simulation realism while reducing the solution space.

7.1.1 Materials and Methods

The designer of a photogrammetric camera network is faced with, at least, three practical problems:

- representing the geometry of the object or scene in a compact, yet not too coarse, fashion
- creating one or more initial imaging configurations which are considered optimal or econo-

mical for the task

- choosing the criteria that will guide the process of selecting or optimising one of the initial configurations

The track surface itself could be decomposed into segments of canonical geometry with varying width, height, and curvature in order to independently study each segment according to its geometric type. However, in a non-signalised photogrammetric survey the object-space point configuration is site-specific, which means that for the parts of the track that are not covered by roofs or blinds, additional feature points will be matched on the surrounding environment. One option would be to use a static geometric representation of the track-site scene; for example sampling 3D coordinates on a half-pipe or open rectangle, q.v. figure 2.3a. On the one hand, this would imply that the same object-space point distribution was expected to be used on-site. On the other hand, it would require pseudo-random or deterministic assumptions about the image projections of each pre-defined point. Another option could be that of using coded targets at randomised positions, so that whether or not a target point is projected on one of the images in the assessed imaging configuration would be defined from objective geometric criteria related to the template matching algorithm; however, this option would override the feature point correspondence problem.

Instead of directly using discrete point coordinates as an abstraction of the track site, images of real scenes or rendered views of virtual scenes were used so that the image correspondence problem, feature point distribution, and the feature localisation accuracy could be implicitly taken into account in a more realistic fashion. Nevertheless, no attempt was made to simulate the reflective properties of ice, or any other surface type for that matter. The question was, with all other attributes being equal and favourable, how does the geometry of the imaging configuration affect the reconstruction result?

Scene Representation The first experiment was conducted on an abstraction of the track's surface—including roofs and blinds—namely a cylindrical surface, which was simple enough to allow for the automatic processing and assessment of a large number of experiments, yet representative of its geometry, q.v. section 7.1.2. For the second experiment, a subset of the previously assessed configurations was used in order to render views of a 3D model of a 110 m segment of the Igls track, q.v. section 7.1.3. Finally, an attempt to verify the results was made by using real images captured on the same track segment, q.v. section 7.1.4.

In the first two experiments, realistic textures with small ground sampling distance were applied to the 3D models. The model of the track segment was created by an image-based scanning

process, so a texture map was generated from the acquired images. Textures of materials that are commonly used in construction sites were applied on the cylinder in order produce generally applicable results, as at that point the problems related to the reflectance characteristics of ice and how they may influence the photogrammetric process were not examined. All surfaces were assumed to have Lambertian reflectance and no attempt was made to recreate realistic lighting conditions. As the effects of light interaction with glossy surfaces depends on the geometry of the scene and are hard to simulate, taking it into account makes the problem potentially intractable and the results worthless. On the other hand, it is also computationally demanding and requires significantly longer rendering times. Ignoring this effect allowed for the assessment of each camera network on the merits of its geometric properties alone. Case studies on the full length of real tracks with glossy ice surfaces are presented in chapters 10 and 11.

Imaging Configurations The camera networks under assessment were described with respect to an average smooth camera trajectory—called the reference trajectory—which in turn was defined with respect to the track's centreline alignment. In theory, this alignment can be considered to be the curve that is parallel to the track's sidewalls; equidistant from them along the straight segments and at a distance of approximately 0.7 m from the inner wall at the bends.

Three vertical positions—high, middle, and low—were considered at fixed heights above the floor, corresponding to the standing, bending, and kneeling postures, respectively, of a camera operator of average height, q.v. figure 7.3a. The camera could be either on, above, or below the reference trajectory or at a lateral distance dY on either side—called the half-base—which is constrained in practice by the available walking space on the track's floor. The width of the flat part of the track is reduced from approximately 1.4 m along the straight segments to less than 1 m along the bends; however, there is certain flexibility in placing the camera towards the almost vertical part of the track's surface along the bends, as the limiting factor is the visibility of the inside wall, as well. Three values were examined for the half-base, namely $dY \in 0.25, 0.40, 0.55m$, leading to an across-track stereo-bases of 0.5 m, 0.8 m, and 1.1 m, respectively.

The orientation of each camera was defined with respect to a reference camera moving along the reference trajectory. The roll angle for all cameras was assumed to be 0°, meaning that the X-axis of the cameras' coordinate system, q.v. figure 3.5, was horizontal and approximately parallel to the track floor's cross section at that point. In order to limit the range of the pitch and yaw angles for the generation of candidate imaging configurations, the results of the preliminary trials on the Whistler and Winterberg tracks were studied, q.v. figures A.1 and A.2, bearing in mind that the estimation of the exterior orientation parameters were the product of a free self-calibrating bundle

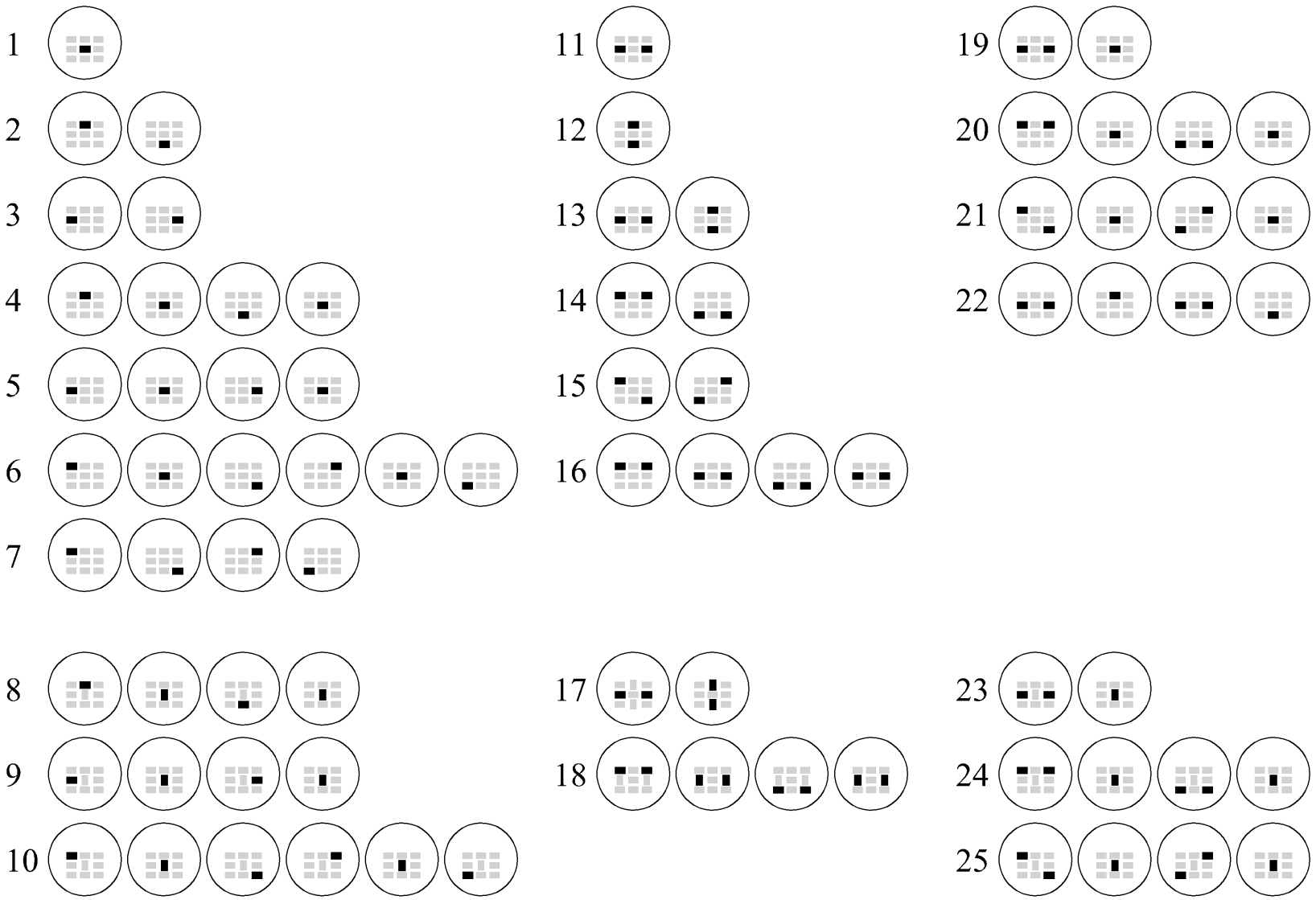
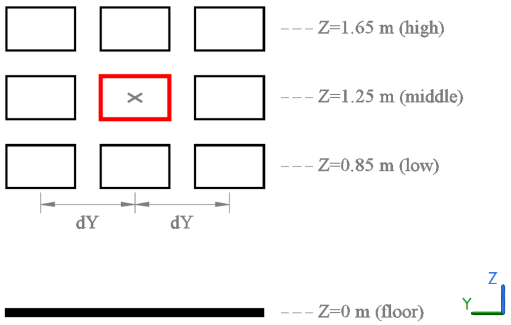
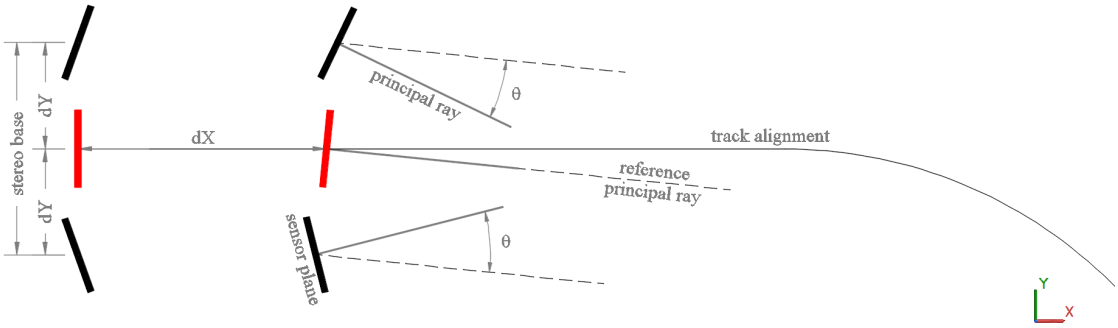


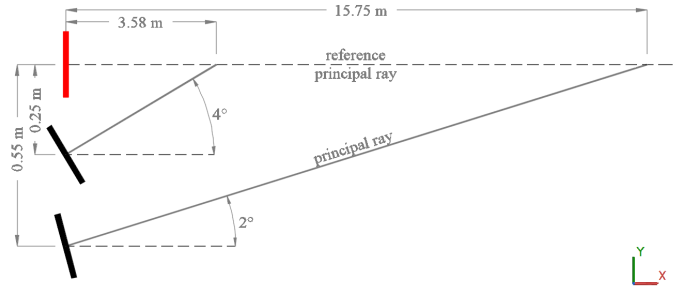
Figure 7.2: Schematic representation of the imaging configurations for the 25 generic blocks under investigation. The three rows at the bottom comprise blocks that include cameras with roll angle equal to 90°.



(a) Back view. The trajectory of the reference camera is marked with a cross.



(b) Top view.



(c) The value range for the yaw rotation of the off-centre cameras was based on a notion of convergence where the principal rays would intersect with the reference path within a working range greater than 2 m.

Figure 7.3: Schematic representations of the imaging blocks geometric configuration. The reference camera is highlighted in red.

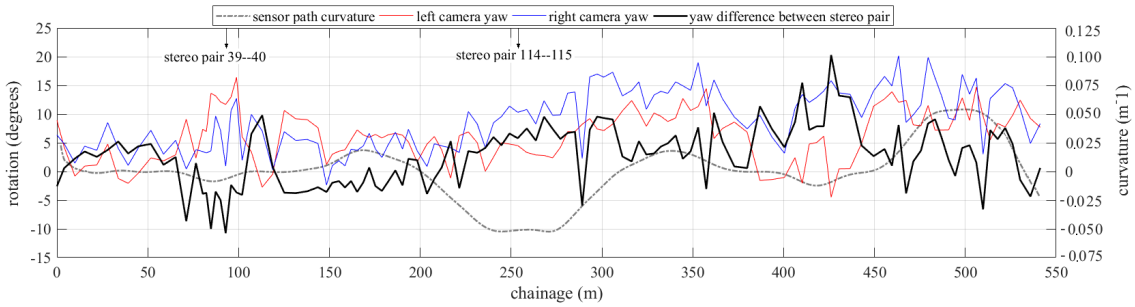


Figure 7.4: The yaw rotation of the left and right cameras of the stereo pairs, which were collected along a segment of the Winterberg track, with respect to the track's centreline alignment tangent. A positive difference indicates a convergent pair. Positive curvature indicates a left turn.

adjustment—therefore, potentially imprecise—and that no instructions were given with respect to the convergence properties of the stereo-pairs which were collected along the Winterberg track.

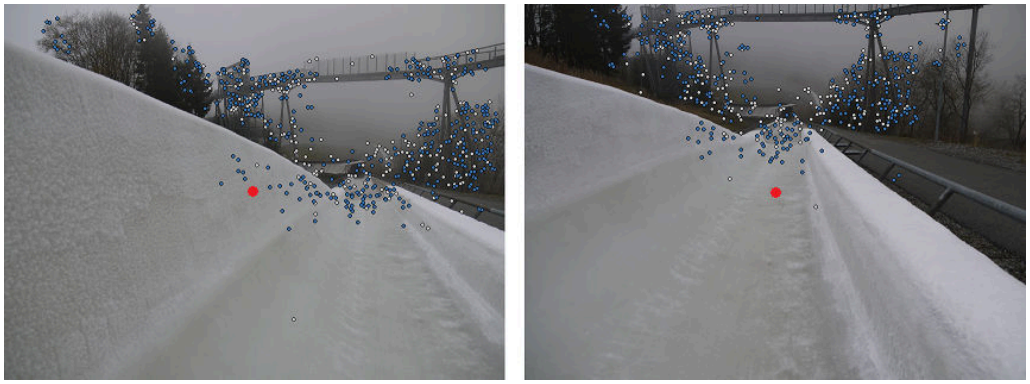
In general, the principal ray of the reference camera is not necessarily parallel to the alignment's tangent at that station, q.v. figure 7.3b, as it relies on photographic composition which is specific to the site. The yaw rotation of the left and right cameras of the stereo pairs, which were collected along a segment of the Winterberg track, with respect to the track's centreline alignment tangent are plotted against its curvature in figure 7.4. Contrary to expectations, there is no clear correlation between the yaw angle of each camera and the curvature sign of the track segment ahead

It is also apparent that it is hard to precisely control the orientation of a hand-held camera. In figure 7.5a, for example, even though there is a clear view of the track ahead, the left camera is pointed to the left; as a result, no features were matched on the right side of the right view. In figure 7.5c, where the right turn is blocked by a blind, the photographer chose to keep the track surface close to the camera in the field of view. It also appears that pointing the camera at the track floor—or roof—right in front of it does not necessarily produce more feature matches.

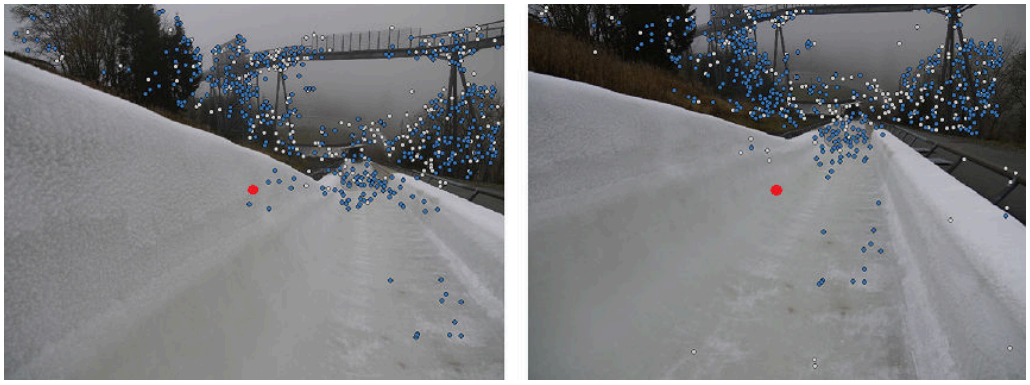
Therefore, the yaw rotation of the off-centre cameras were set to a moderately convergent region of $\theta \in 0, 2, 4°$, q.v. figures 7.3c and 7.3b, so that, in combination with the achievable stereo base, they could provide a field of view centred at distances along the track's centreline that maximise the projection of the track on the overlapping area.[1] In order to strike a balance between large convergence angles and an optimal field of view, the pitch angle for all cameras—with respect to the corresponding reference camera coordinate system—was set to $0°$ for the simulations.

Starting with the aforementioned generic imaging configuration, the main challenge was to

[1]Positive rotation is counter-clockwise following a right-handed coordinate system.



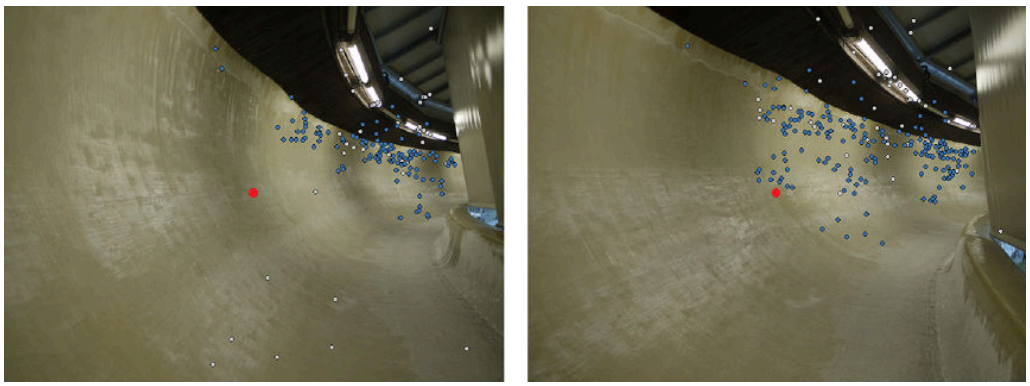
(a) Stereo pair 39–40.



(b) Stereo pair 41–42.



(c) Stereo pair 114–115.



(d) Stereo pair 116–117.

Figure 7.5: Two diverging stereo pairs collected at the Winterberg track.
(●) principal point, (○) detected features, (●) matched feature

define the number of images at each station along the alignment, how long each step between the stations should be, and the configuration of the image block at each station in order to generate the networks that will be assessed. The total number of images at each station and the common interval length dX between successive stations were defined with respect to the total number of images that can be captured along the competition length of a track—which is on average 1400 m—within less than 2 hours. Given the estimation that the time it takes to capture a single image is 6.5 seconds on average, including moving between stations and camera positions, 41–42 images would be required to cover a 60 m long segment, which is equivalent to 1 hour and 45 minutes for the entire length of a typical track and corresponds to 1 image every 1.5 m, 2 images every 3 m, 3 images every 4.5 m, and so on.

The 25 generic imaging configurations that were assessed are depicted in figure 7.2. The leftmost column shows the generic blocks 1–10 with one camera at each step and the middle column shows the generic blocks 11–18 with two cameras at each step. Blocks 19–25 represent alternating one- and two-image blocks 2.25 m apart, as the 4.5 m was deemed too long an interval and thus was split in half. This decision was based on the results presented in section 5.2, q.v. figure 5.5, which indicated that stereo-intervals greater than 4 m could cause the image matching process to fail and the image sequence to break. As the exact geometry of any configuration designed in the office cannot possibly be exactly replicated on site—unless special equipment and plenty of time is available—and in order to slightly depart from critical motions, q.v. section 3.3.1, random noise was added to the exterior orientation parameters of each camera. As a result, images taken from the same camera station but for a different block were rendered independently.

7.1.2 Simulations on a Cylindrical Surface

For the first experiment, the track's geometry was represented by an 80 m long cylinder with its axis parallel to the reference system's X axis, q.v. figure 7.7a. The reference camera was moving along the axis with its principal ray parallel to it. The cylinder's radius was selected equal to 1.5 m based on the orthogonal distances from the feature points that were matched across the image sequences of the Whistler and Winterberg tracks to the sensor path, q.v. figures A.1 and A.2. The texture that was applied on the smooth cylinder surface, which was assumed to be diffusely scattering, had a ground sampling distance of approximately 1 mm. In order to keep this first experiment as general as possible, the texture map comprised high resolution images of concrete floors and brick walls, q.v. figure 7.6. The image sets were created by generating virtual cameras for all the candidate imaging configurations and rendering views in Autodesk 3ds Max (2015).

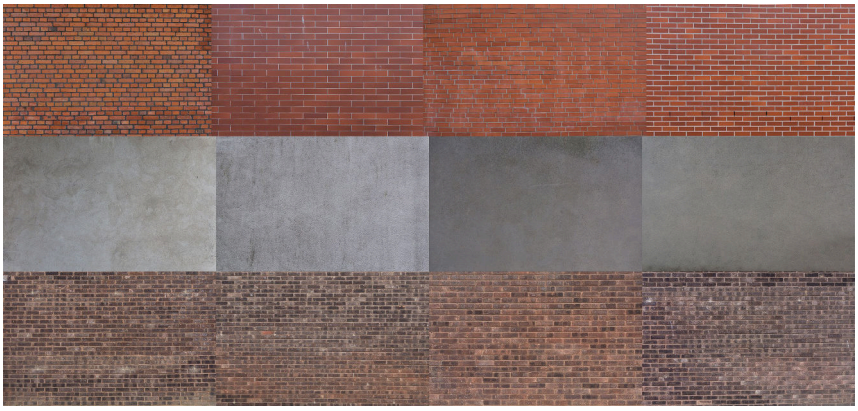


Figure 7.6: One of the four 20000×9420 pixels texture maps that was applied on the cylinder object which was used in the first set of the network design simulations; adapted from ©Texture Library (2013) and ©Texturify (2014).

Table 7.1: The attributes of the imaging configurations that were used in the network design simulations experiments on a cylindrical surface.

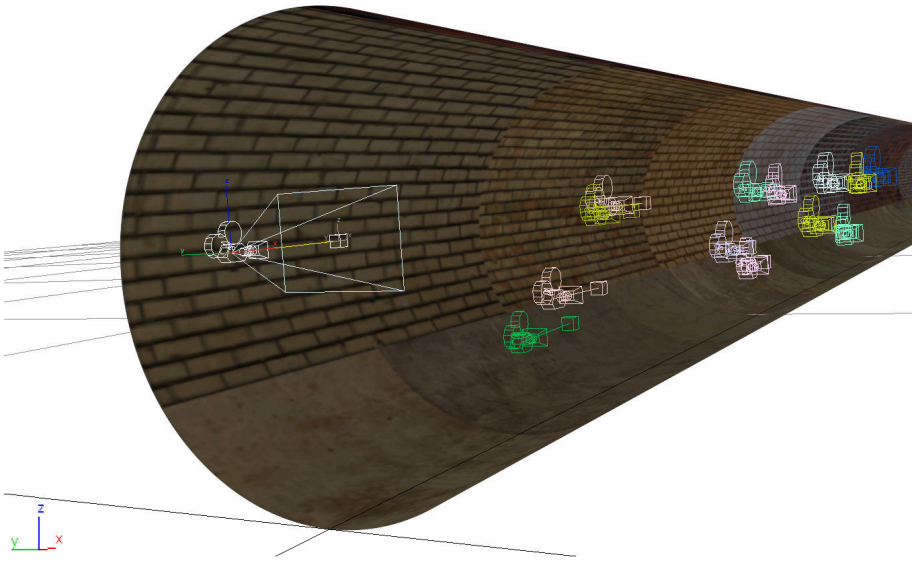
Interval	1 m	2.25 m	3 m
Block ID	1–10	19–25	11–18
Base	0, 0.50[a], 0.80[a], 1.10[a] m		
Yaw	0, 2[a], 4[a] degrees		
Frame Size	4096×2160, 4752×3168, 5616×3744 pixels		
Principal Distance	20, 28, 35 m		

[a] not applicable to blocks 1, 2, 4, 8, 12.

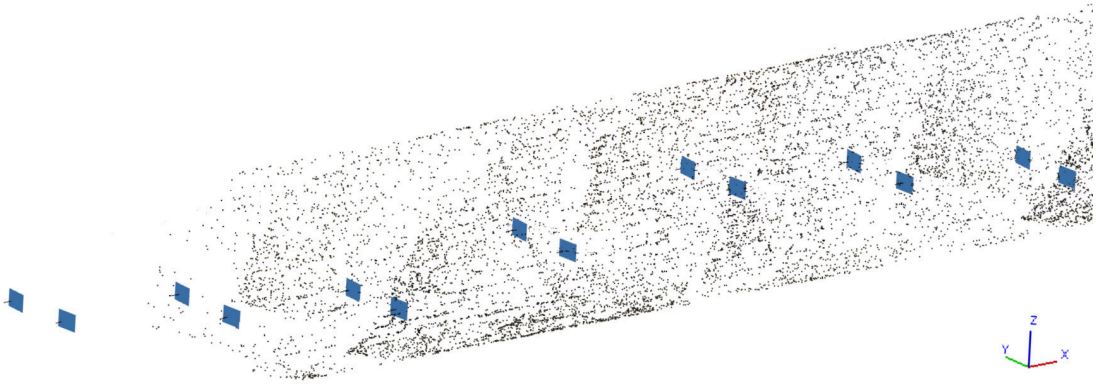
The principal point coordinates, as well as the tangential and radial distortion coefficients, were set equal to zero and no artificial noise were added to the images. Nine sets were created for each network with principal distances equal to 20 mm, 28 mm, and 35 mm at resolutions of 9 MP, 12 MP, and 21 MP using a full frame sensor ($36mm \times 24mm$); the frame sizes were 4096×2160, 4752×3168, and 5616×3744 pixels, respectively.

7.1.2.1 Processing

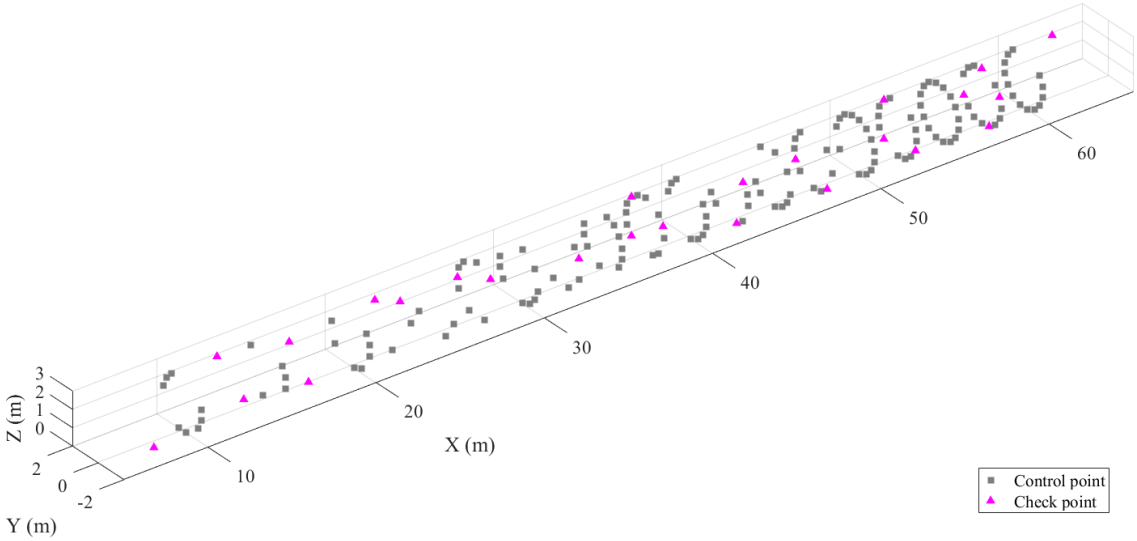
In total, 1665 camera networks—all combinations between the values of each of the design parameters under consideration for every block configuration, q.v. table 7.1—were batch-processed in a common workflow using PhotoScan's Python (1991) application programming interface (API). After the initial image orientation process, three different strategies were used for the bundle adjustment of each network with respect to the definition of the exterior and interior orientation



(a) A perspective view of the textured cylinder used for the network design simulations.



(b) A sparse point cloud generated in PhotoScan using block 16.



(c) The positions of the coded targets that were used as control and points.

Figure 7.7: The test object that was used in the first set of camera network design simulations.

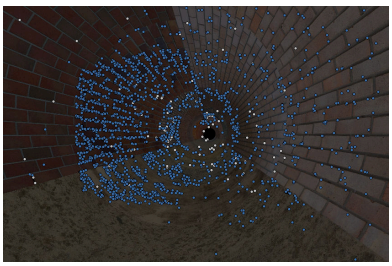
parameter sets. In the beginning, a reference dataset was created by fixing the exterior and interior orientation of the cameras to the known values that were used to generate the images. A second set was created by a free bundle adjustment, assuming a pre-calibrated camera with fixed interior orientation, and, lastly, a free self-calibrated bundle adjustment where both exterior and interior orientation were unconstrained. The adjustment for each network was performed in an iterative fashion until all points with a reprojection error greater than 0.5 pixels were removed.

7.1.2.2 Assessment

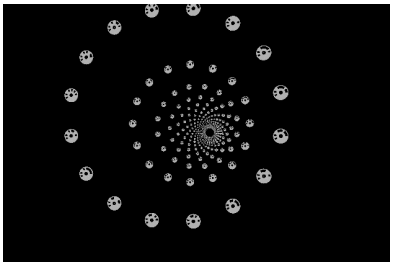
The intent of this set of simulations was to compare weakly convergent imaging configurations with respect to the reconstruction of a narrow space under favourable radiometric conditions. With all the other influences being equal, the triangulation process should only be affected by the geometry of the converging rays connecting the feature matches in object, sensor, and image space under the varying perspective distortion of oblique images. A reduction of the number of candidate networks was sought, so that a more thorough investigation on a compact subset could follow, q.v. section 7.1.3.

The use of a cylinder as a test object allowed for the comparison of the object-space coordinates of every single feature point to a single reference value, namely its nominal radius. Using a cylindrical surface, instead of any other arbitrary surface with known geometry, was based on the premise that it would enable the assessment of image-based scanning of non-signalised scenes involving low incidence angles. Nevertheless, using the radial distance alone as a quality measure could bias the results, as the cameras' principal ray is almost parallel to the cylinder's axis, thus, depth uncertainty—errors along the reference system's X axis—could appear as smaller errors along the cylinder's surface normal. This was one of the reasons why fitting a cylinder to the sparse point cloud in order to compare the estimated to reference cylinder radius was avoided. The other reason was that, in a targetless scene, each photogrammetric model was referenced in a different coordinate system. As it was not possible to predict the features that will be detected on an image, in order to generate reference coordinates from the 3D model, the alternative of using the known exterior orientation of the cameras would induce registration errors due to the different camera networks and projective compensation.

To overcome this problem, the images of the cylinder texture were replaced with images of coded targets spread along the cylinder's surface, q.v. figure 7.8, in order to detect them and triangulate their positions using the already calculated cameras' exterior and interior orientation from the solution based on feature matching. These images were created by rendering views of



(a) The textured model of the cylinder. The detected (white) and matched (blue) feature points are also visible.



(b) The coded target array from the same viewpoint.

Figure 7.8: Rendered views of the cylinder that was used in the network design simulations.

the target array using the same virtual cameras that generated the original image set, q.v. figure 7.8a. The registration of all the sparse point clouds, comprising the generated feature points, to the same reference coordinates system was based on the coordinates of 25 of the targets, commonly detected in all networks, q.v. figure 7.7c. The assessment of each network for each processing strategy was based on the total root-mean-squared error (RMSE) of the radial distance from each point in the sparse point cloud to the X axis and the total RMSE of the coded target coordinates that were not used for the registration, q.v. equation 7.1. In both cases, points or targets with residuals greater than 5 m were considered blunders and were not taken into account during the assessment, leaving an average 43500 points and 150 targets in each network.

$$RMS\,E_{total} = \sqrt{\frac{\sum_{i=1}^{n}(X_i^{est} - X_i^{ref})^2 + \sum_{i=1}^{n}(Y_i^{est} - Y_i^{ref})^2 + \sum_{i=1}^{n}(Z_i^{est} - Z_i^{ref})^2}{n}} \tag{7.1}$$

The results of the self-calibrating bundle adjustments, grouped according to the values of each one of the factors under investigation, are summarised using boxplots.[2] The first observation on the results from the self-calibrating bundle adjustments results on feature point and target coordinates RMSE which are shown in figures 7.9 and 7.10, respectively, was that blocks 1, 2, 4, 8, and 12—which comprise cameras positioned on, above, or below the sensor path, therefore have across-track stereo base equal to zero—appear to be among the worst performing ones. The scatter plots and histograms of the results, q.v. figures B.1 and B.2, show that the inclusion of cameras with varying roll angles in the configurations only affects the object space accuracy of weakly convergent networks, i.e. those that include yaw angles and stereo bases close to zero. This effect

[2]On each blue box, the central red line indicates the median; the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The notches on either side of the red line express the 95% confidence interval for the median. Black whiskers extend 1.5 times the interquartile range away from the top and bottom of the box; points that fall outside are marked with a red cross.

is multiplied by the linear nature of the test object, as well. The results are consistent to those from the reference and pre-calibrated bundle adjustments, where the positive effect of higher image resolution is even more evident. A rigorous statistical analysis of variance (ANOVA) was not pursued as there was no indication that its fundamental assumptions hold for the datasets under consideration.

7.1.3 Simulations on the 3D Model of a Track Segment

Fourteen generic blocks—namely blocks 3, 5, 6, 7, 10, 11, 13, 14, 16, 17 ,18, 20, 22, and 24—from the previous experiment were used in a second set of simulations with the sensor moving along a curve. One more imaging configuration, comprising block 5 along the straight segments and block 1 along the bends, was included and will be referred to as block 26.

The geometry of the scene was represented by a 3D mesh of a 110 m long segment of the Igls track in the iced condition, q.v. figure 7.11; more specifically, bend 3 including the entrance and exit straights, q.v. figures 7.11b and 7.11c. The 3D mesh was created in PhotoScan using a total of 486 images with 21 MP resolution collected in 3×3 blocks at $dX = 1.75$ m intervals; the lateral distance dY for the off-centre cameras was approximately 0.4 m. The images were acquired using a Canon EOS5D Mark II DSLR camera with a Canon EF 28mm f/1.8 USM prime lens, q.v. table D.1. The free self-calibrating bundle adjustment RMS reprojection error was 0.5 pixels. The dense point cloud, which was used to reconstruct the surfaces of the track and the surrounding environment, was generated from the entire image set at 1/16 of their resolution, comprised 70 million points and provided full track coverage. Its scale and orientation was based on control point coordinates measured on site, q.v. table E.2. The texture map that was applied on the mesh had a ground sampling distance of approximately 5 mm, q.v. figure 7.11e.

The imaging networks were generated with respect to a reference alignment that was 1.25 m above the track floor and parallel to the inner wall at a distance of 0.7 m. The stereo base was set to 0.65 m ($dY = 0.325m$) and the yaw angle was set to 3° with respect to a reference camera with its principal ray parallel to the alignment's tangent at each camera station. The views were rendered according to the process that was described in section 7.1.2, using a full frame virtual camera with a resolution of 5616×3744 pixels and a moderately wide virtual lens with a principal distance of 28 mm.

The quality assessment of the bundle adjustment was based on virtual coded targets which were added in the scene; three at the start and nine at the end of the track segment. The former were used to register all networks in the same reference system and the latter to estimate the 3D

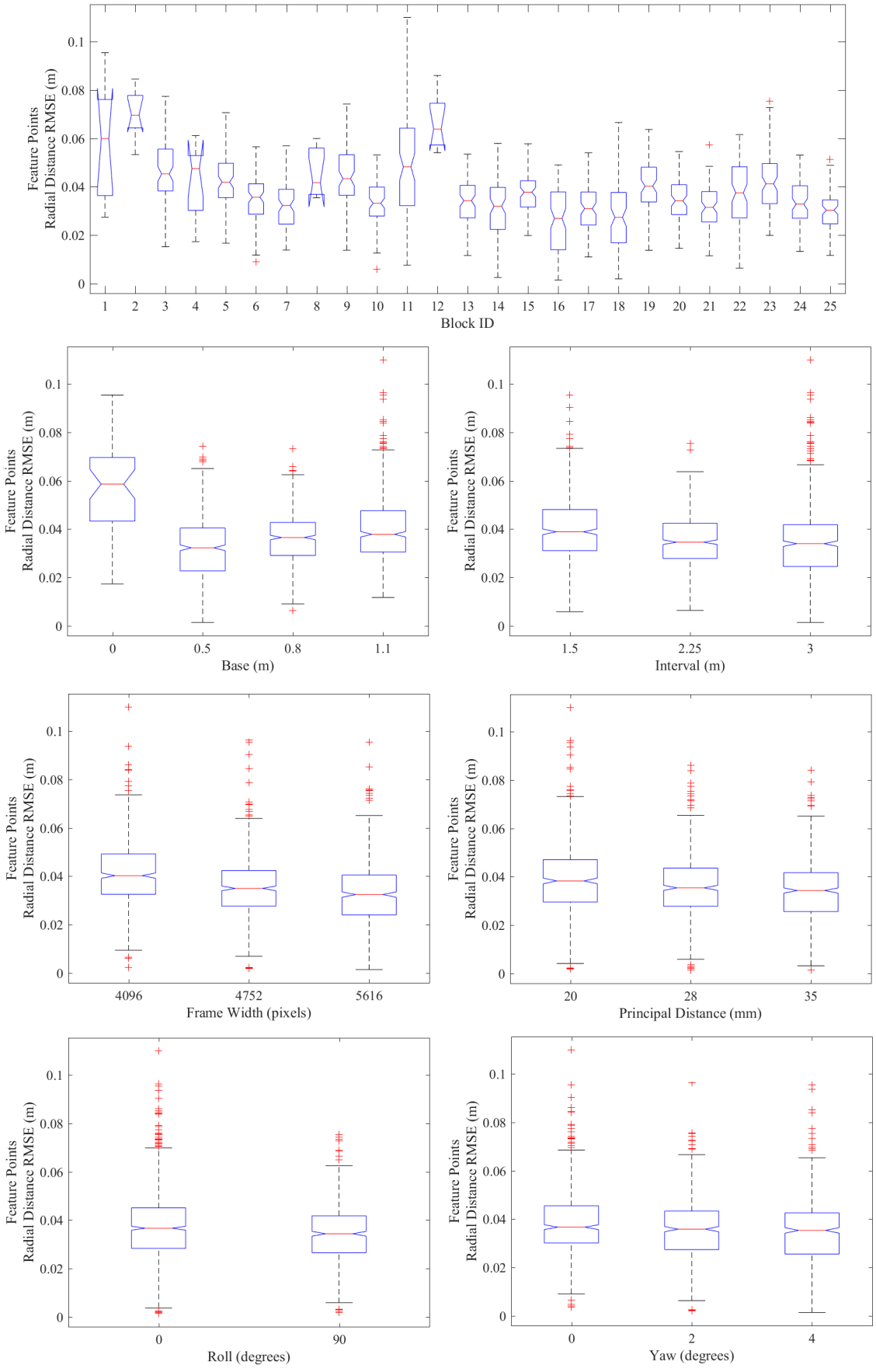


Figure 7.9: Boxplots of the total RMS of the feature points' radial distance errors, which were estimated from the self-calibrating bundle adjustment simulations.

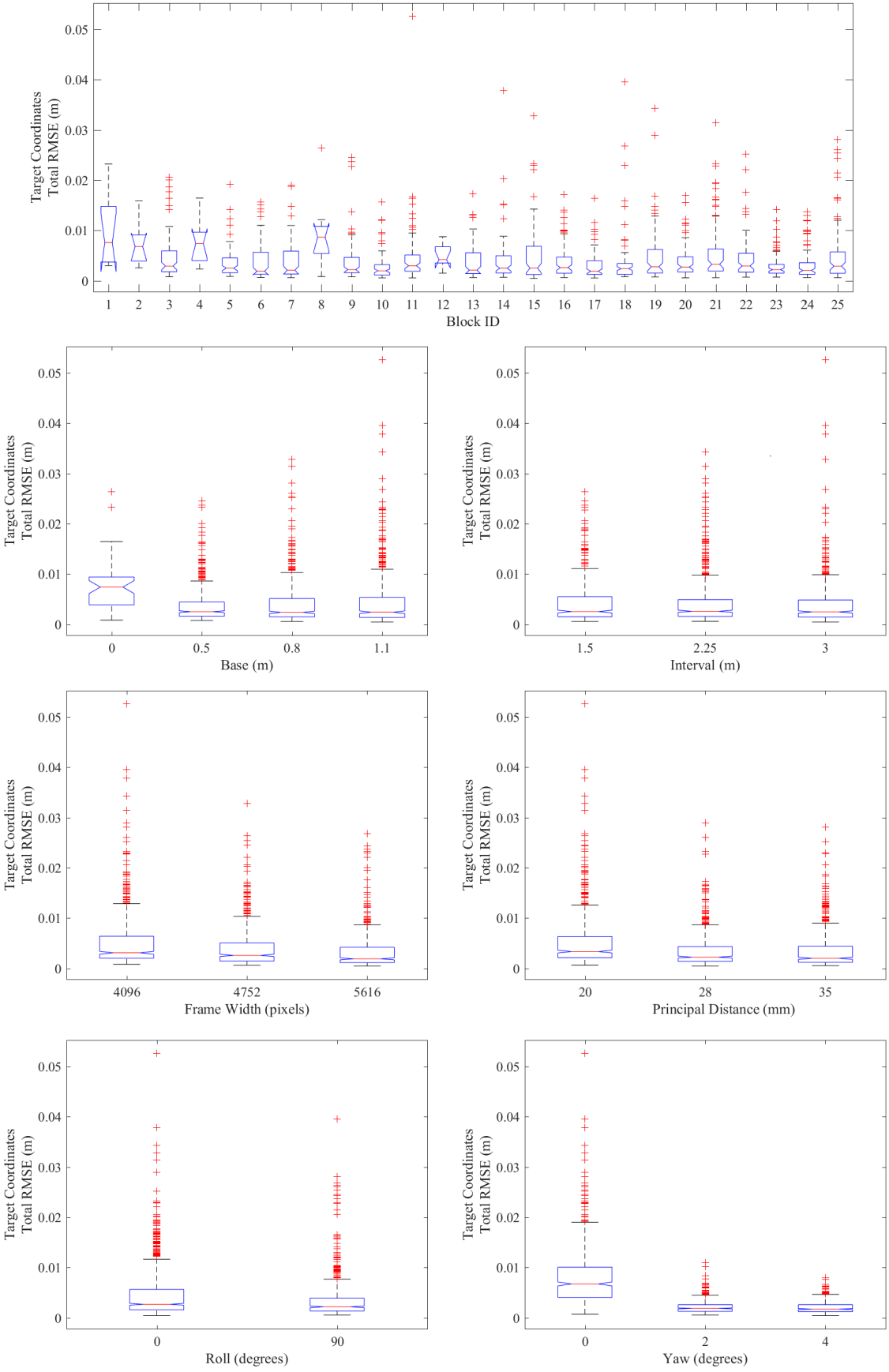
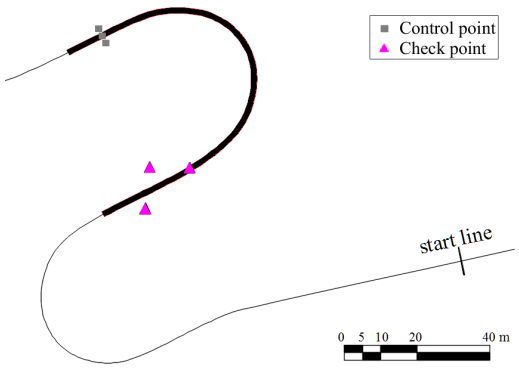
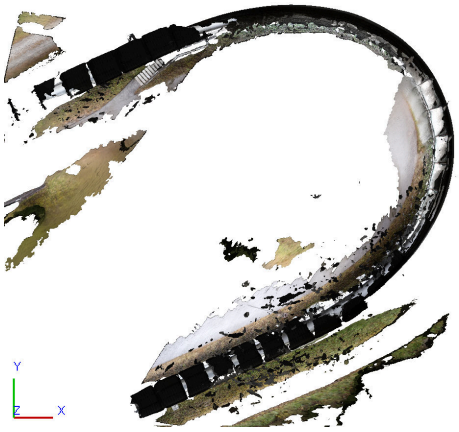


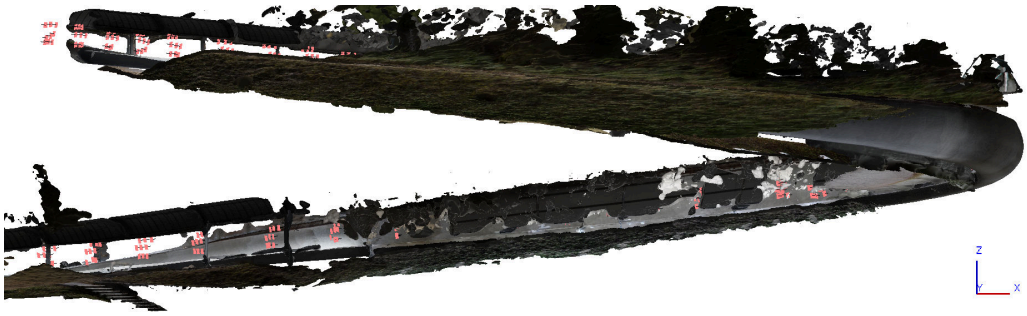
Figure 7.10: Boxplots of the total RMSE of the coded target coordinates, which were estimated from the self-calibrating bundle adjustment simulations.



(a) The centreline of the Igls track along the first three bends; the segment under consideration is highlighted.



(b) Top view of the textured 3D mesh created along corner 3.



(c) Side view of the textured 3D mesh; the image blocks are also visible in red colour.



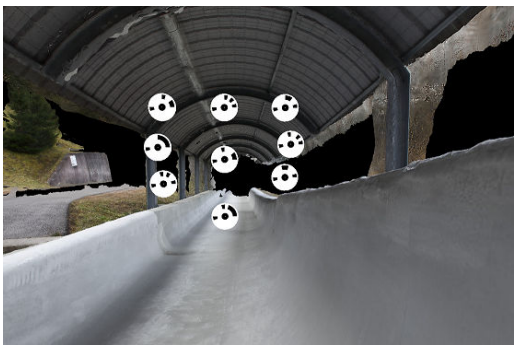
(d) One of the images that were used to create the 3D mesh of the track.



(e) A perspective view of the textured 3D mesh from the viewpoint of the image in figure 7.11d.



(f) One of the four rendered images of the control point targets.



(g) One of the four rendered images of the check point targets.

Figure 7.11: The test scene that was used in the second network design simulation experiment.

Table 7.2: Parameter settings for the image orientation simulations on the 3D mesh of bend 3 at the Igls track.

Interior Orientation Parameters							
c pixels	x_{pp} pixels	y_{pp} pixels	K_1	K_2	K_3	P_1	P_1
4368	0	0	0	0	0	0	0

Processing Parameter Settings				
Quality	Tie Point Limit	Match Point Limit	Adaptive Camera Model	Pair Pre-selection
Highest	60000	4000	No	Yes

drift of the reconstruction, q.v. figure 7.11a. In order to get comparable results, the triangulation of the coded targets was based on an additional block of four images—one at either end—which was common for all networks, q.v. figures 7.11f and 7.11g. These eight images were aligned with the rest of the images based on feature correspondences.

The image orientation and free self-calibrating bundle adjustment was performed in PhotoScan using the settings and initial parameter values of table 7.2. Ten trials were conducted on each set of images, in order to assess the effect of the random feature detection process. The total RMS reprojection errors were less than 1/3 of a pixel for all adjustments. After each adjustment, the coded targets were automatically detected on the images and a similarity transformation was calculated based on the control points. The registration errors for all networks and the total RMSE at the check points are shown in figure 7.12. The configurations with the highest and lowest RMSE—namely 11 and 13, respectively—belong to the blocks that have two images at each step. It appears that the blocks can be separated in two sets with respect to whether or not the mean RMSE between the trials is lower than 10 mm.

7.1.4 Simulations on a Track Segment

Blocks 5, 11, and 26 were selected from the previous experiment, as the simplest among the best- and worst-performing—according to the results in table 7.3—in order to assess them with respect to the accuracy and completeness of the surface reconstruction of a track segment. The images for all three imaging configurations originated from the 486 images that were collected on the Igls track along bend 3 and were used for the reconstruction of the 3D model which served as the reference object for the experiments that were presented in section 7.1.3. The first 80 m of that

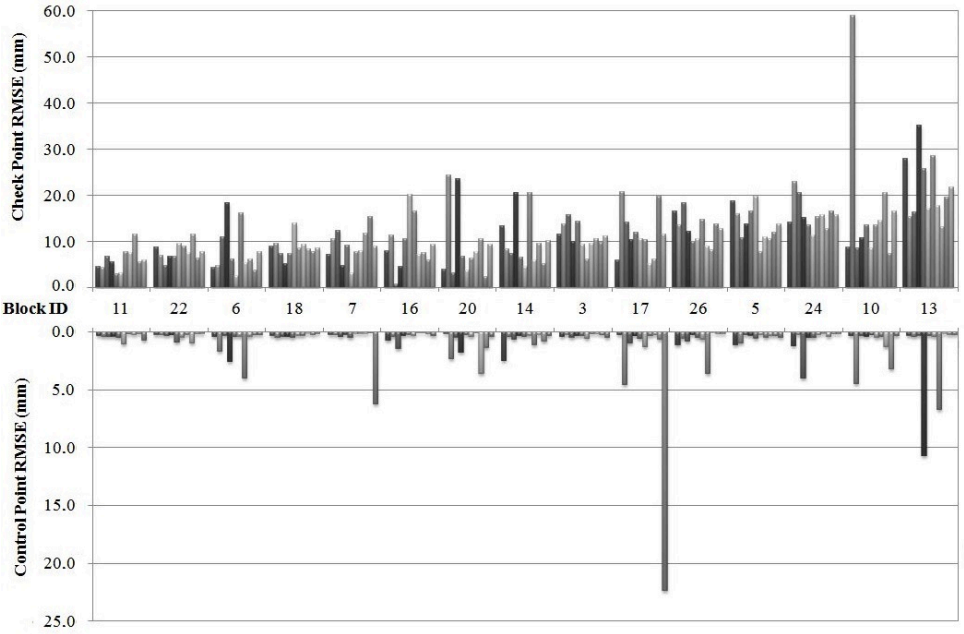
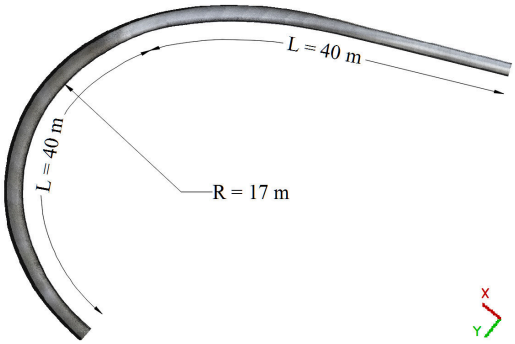


Figure 7.12: The total RMSE of the check point coordinates for the camera network design simulations that were conducted on the 3D model of bend 3 of the Igls track. The blocks are ranked according to the arithmetic mean of the check point RMSE between all trials.

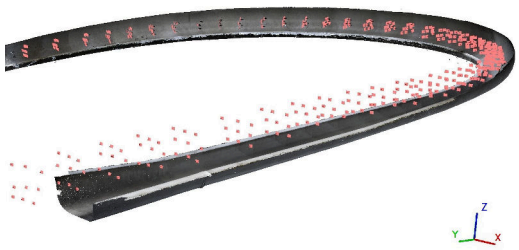
model—split equally between the straight entrance to the bend, including the transition segment, and the circular arc with a 17 m radius—served as the reference object for the assessment. The along-track interval was on average 3.5 m for block 11 and 1.75 m for blocks 5 and 26; the stereo base was approximately 0.8 m (half-base $dY \approx 0.4$ m) for all 3 blocks. In addition, a network comprising block 11 with $dY \approx 0.2$ m was also considered. The generic blocks that comprise images at the bending and kneeling positions may increase the total acquisition time, as they are more physically demanding; in addition, these postures are more unstable and may result in blurry images. Therefore, two variations were examined for the networks under assessment: one at the bending position at 1.25 m above the track floor and one at the more comfortable standing position at a height of 1.65 m.

In essence, the performance of a very dense network was compared to 8 reduced versions of itself. Even though all 8 networks under assessment were created using subsets of the same image set, the image detection and matching process was performed independently for each subset, comprising 54 images, using an upper limit for feature detection and matching of 100000 and 4000 points, respectively. The total RMS reprojection error of the self-calibrating bundle adjustment was less than 0.5 pixels for all networks. The dense point clouds were generated using the image at 1/4 of their resolution and a mesh was created for each one, comprising 1 3D face for every

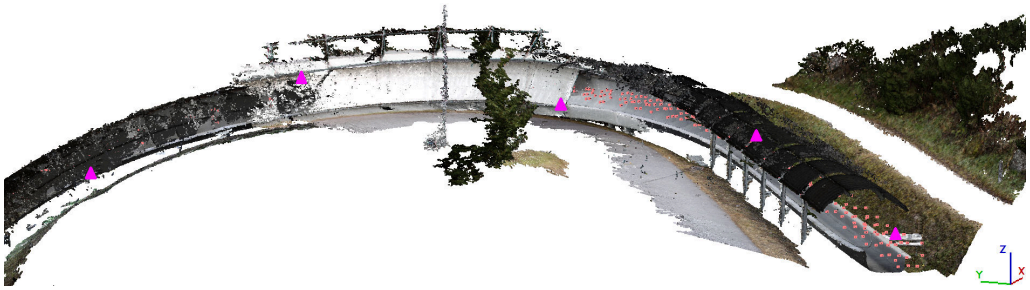
10 points leading to a total number of 2.5–4 million faces. The 3D models were registered to the reference coordinate system using 5 control points that were measured manually on the images. All networks shared the same control point observations on the common images—their total RMS reprojection error was less than 0.33 pixels. The RMSE of the similarity transformation to the reference coordinate system, the completeness of the reconstructed surface, and the maximum distance of the closest 95% of the dense point cloud points to the reference surface are summarised in table 7.3 and figures 7.14–7.16.



(a) Top view of the track's surface that is covered in ice.



(b) The 54 3×3 image blocks.



(c) The control points marked with magenta triangles on a perpsective view of the dense point cloud. Their positions were selected at 20 m intervals along the track and measured on the images in order to register the networks under assessment in a common coordinate system.

Figure 7.13: The 3D model of bend 3 of the Igls track.

In contrast to the results that were reported in section 7.1.3, the performance of block 11 for dense reconstruction appears to be inconsistent and worse than that of block 5 at the middle position, which is similar to that of block 26 at the high position. In fact, block 11 in the high position also has the greatest similarity transformation error which indicates problems in the image orientation process as well. In addition, the completeness of the surfaces that were reconstructed from blocks at the middle position was consistently lower than those at the high position. It can be said that, to a certain extend, the effectiveness of the point cloud filtering process may have masked the surface reconstruction errors.

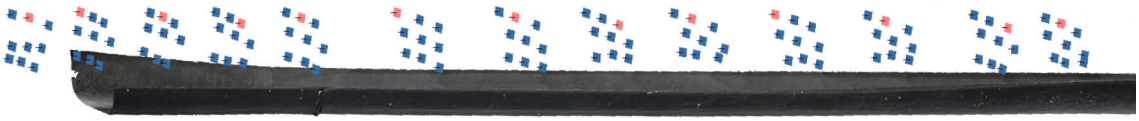
Table 7.3: The RMSE of the similarity transformation to the reference coordinate system, the completeness of the reconstructed surface, and the maximum distance of the closest 95% of the dense point cloud points to the reference surface.

Block ID	Position	Completeness (%)	Dense Point Cloud to Surface Distance 95% percentile (m)	RMSE of the Similarity Transformation to the Reference GCPs (m)
5	high	83	0.046	0.031
5	middle	76	0.040	0.012
11	high	80	0.048	0.030
11	middle	66	0.045	0.026
26	high	82	0.045	0.016
26	middle	74	0.047	0.020
11 out	high	97	0.056	0.048
11 out	middle	85	0.041	0.027

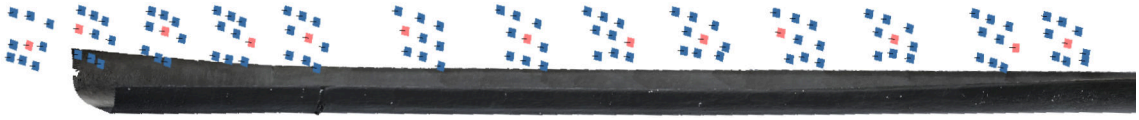
In practice, the sensor's distance from the floor at the standing position, q.v. figure 7.3a, depends on the user's height; in addition, the most appropriate sensor height from the track floor varies along its length, as it may be lower at the entrance or exit of a corner but higher towards the middle, q.v. figure 2.5. Therefore, all 3 blocks, at varying heights above 1.25 m from the track floor, will be considered for the case studies which are presented in part IV.

7.2 Track Image Segmentation

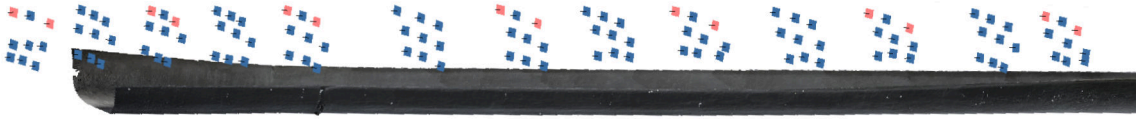
The surface reconstruction process, either from a point cloud or directly from the depth maps which were estimated for each image, q.v. section 3.6.1.1, can be considered as the last stage of the strictly photogrammetric pipeline before proceeding to the 3D geometric modelling process, q.v. figure 4.1, and can potentially be the most time consuming. Given a set of images, the most straightforward way of reducing the processing time is by reducing their resolution. However, as not every part of the track scene needs to be represented at the same level of detail, instead of generating a single dense 3D point cloud or mesh for the entire scene at a lower resolution, it might be more reasonable to process the part of the image where the object of primary interest is depicted. The hypothesis was that performing the dense image matching twice—once to create high quality depth maps only for the part of the image where the track is projected on and once more, at a lower resolution, for the surrounding environment—would be equally efficient without compromising the quality of the result. If the model is only used for trajectory simulations, reconstructing only the track surface as a separate entity from all the other structures around it, allows for better control over the point density; it also means that the appropriate ice texture and reflectance properties can



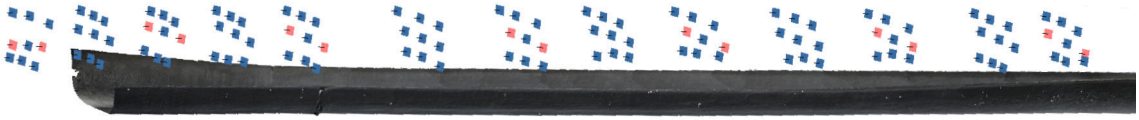
(a) Block 5 - High Position ($dY \approx 0.4m$).



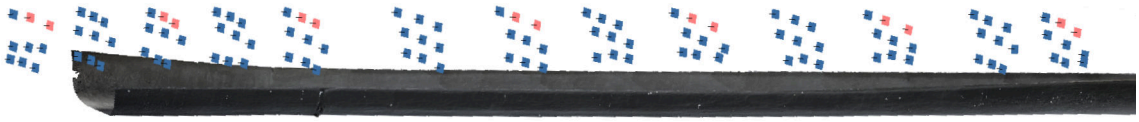
(b) Block 5 - Middle Position ($dY \approx 0.4m$).



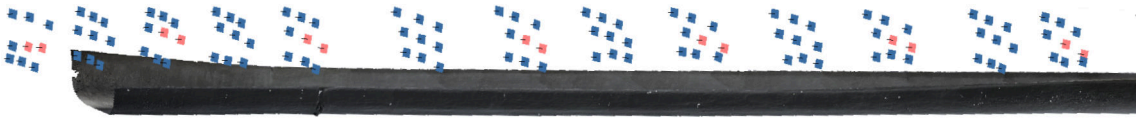
(c) Block 11 - High Position ($dY \approx 0.4m$).



(d) Block 11 - Middle Position ($dY \approx 0.4m$).

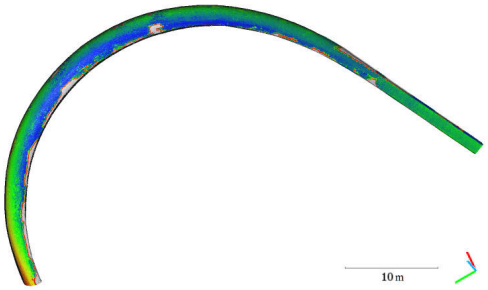


(e) Block 11 - High Position - Outer Path ($dY \approx 0.2m$).

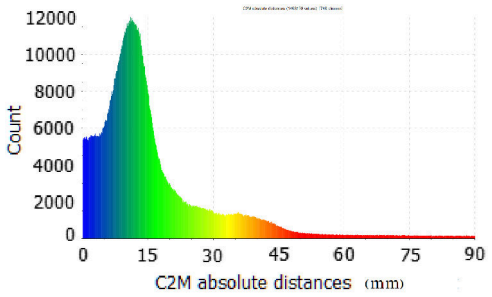


(f) Block 11 - Middle Position - Outer Path ($dY \approx 0.2m$).

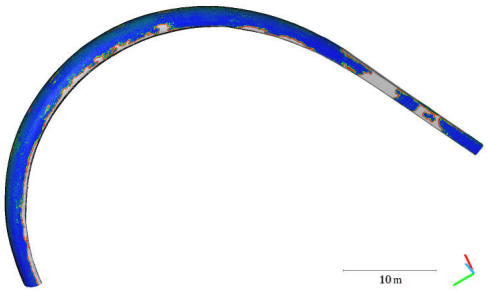
Figure 7.14: The imaging configurations for blocks 5 and 11 that were used for the dense point cloud generation along bend 3 of the Igls track. Block 26 along the straight part of the track comprised the same images as block 5; along the bend, the central image at the corresponding vertical position was used.



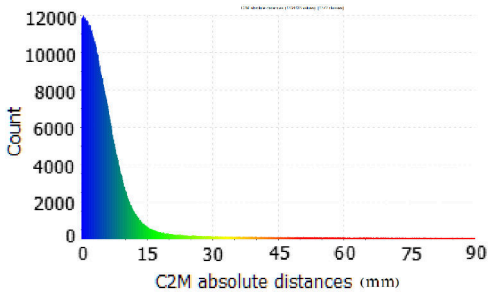
(a) Block 5 - High Position: Dense point cloud. Completeness: 83%



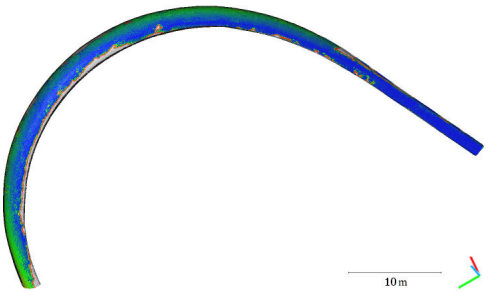
(b) Block 5 - High Position: Histogram. GCP RMSE: 31 mm



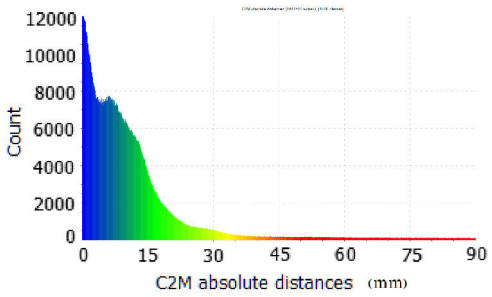
(c) Block 5 - Middle Position: Dense point cloud. Completeness: 76%



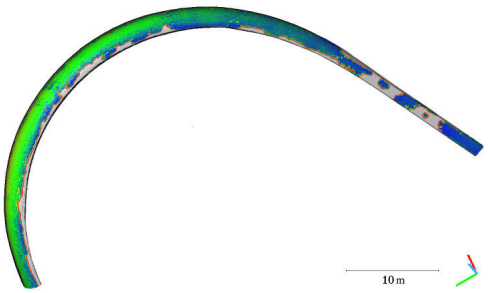
(d) Block 5 - Middle Position: Histogram. GCP RMSE: 12 mm



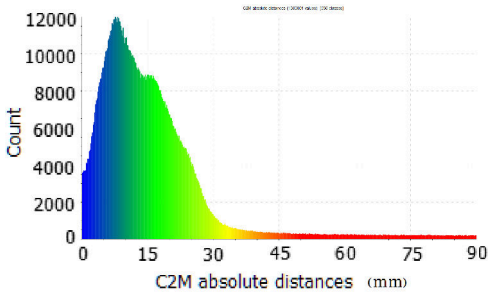
(e) Block 26 - High Position: Dense point cloud. Completeness: 82%



(f) Block 26 - High Position: Histogram. GCP RMSE: 16 mm

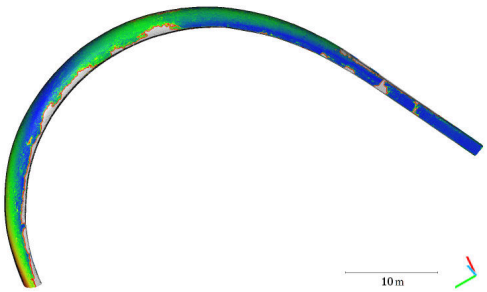


(g) Block 26 - Middle Position: Dense point cloud. Completeness: 74%

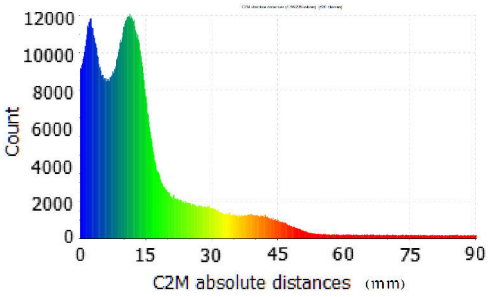


(h) Block 26 - Middle Position: Histogram. GCP RMSE: 20 mm

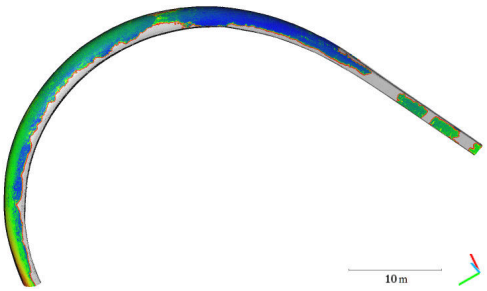
Figure 7.15: Colour-coded dense point clouds and histograms of their distances from the reference object of bend 3 of the Igls track, for each network comprising block 5 or 26.



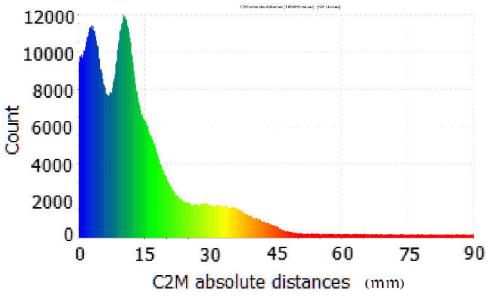
(a) Block 11 - High Position: Dense point cloud. Completeness: 80%



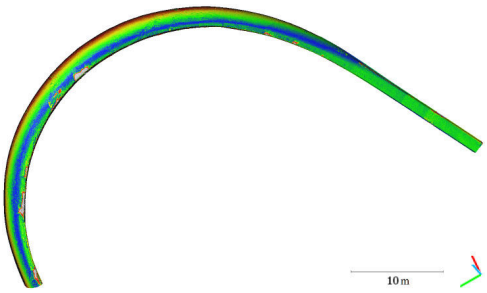
(b) Block 11 - High Position: Histogram. GCP RMSE: 30 mm



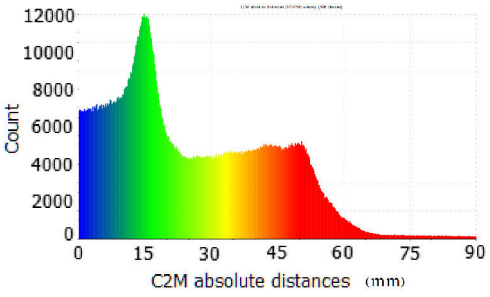
(c) Block 11 - Middle Position: Dense point cloud. Completeness: 66%



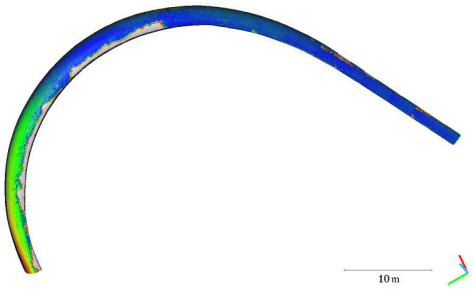
(d) Block 11 - Middle Position: Histogram. GCP RMSE: 26 mm



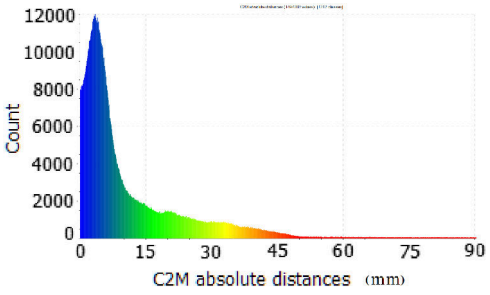
(e) Block 11 - High Position - Outer Path: Dense point cloud. Completeness: 97%



(f) Block 11 - High Position - Outer Path: Histogram. GCP RMSE: 48 mm



(g) Block 11 - Middle Position - Outer Path: Dense point cloud. Completeness: 85%



(h) Block 11 - Middle Position - Outer Path: Histogram. GCP RMSE: 27 mm

Figure 7.16: Colour-coded dense point clouds and histograms of their distances from the reference object of bend 3 of the Igls track, for each network comprising block 11.

also be simulated within the graphics engine of a virtual reality system. The surrounding track environment is also necessary but its use is primarily visual; the presence of a roof, a blind, or a tree at the correct position in the athlete's field of view is what matters most, whereas the fine geometric details are of secondary importance at high speeds and can be conveyed by a high resolution texture applied on a lower resolution 3D mesh.

Creating a mask for each image in order to indicate the part on which the track is projected on requires a binary segmentation of each image to foreground and background, i.e. the track surface and its surrounding environment, respectively. As the number of images that is required for the reconstruction of a track can vary from a few hundred to a few thousand, an automatic workflow for track segmentation was developed. It was built around the interactive "Lazy Snapping" algorithm in combination with SLIC superpixels, q.v. section 3.5.1, so that it can be applied to images of the track covered with ice or of its concrete structure, without explicitly defining a specific material signature. The seed selection for the foreground and background pixels was based on the known image exterior orientation, the sparse point cloud which was generated from the image orientation process, and the projections of each point on the images, thus overriding the need for user-interaction. It is assumed that a curve which approximates the camera's trajectory along the track is available and that the photogrammetric model is approximately scaled and oriented with respect to a local coordinate system; more details on initialising the sensor's path and the datum definition, even without ground control points, are given in section 8.1. It is also assumed that the images are collected in landscape mode from the point of view of the athlete sliding or walking on the track—in the general case, as the exterior orientation is known, the camera roll angles can be taken into account.

Image preprocessing Firstly, all the images are resized so that their width is equal to 1000 pixels and transformed to the HSV colour space in order to perform the track segmentation on the saturation channel, q.v. figures 7.17b and 7.18b. The image is then over-segmented using 2000 SLIC superpixels with their compactness set equal to 1, so that they can retain a canonical shape but also adapt to the colour information and the track's edges.

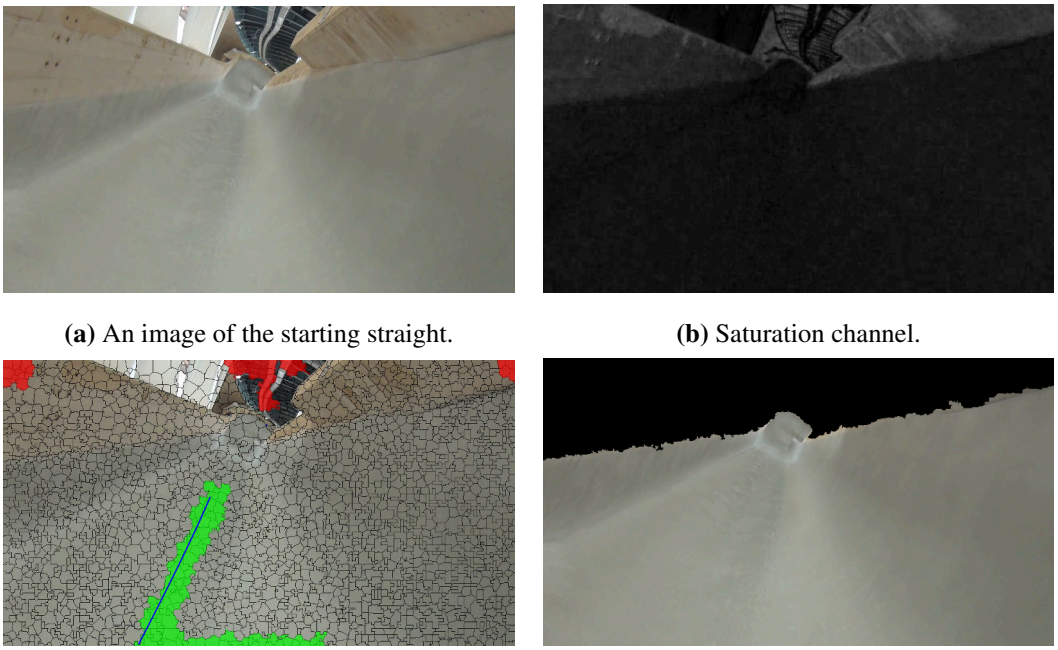
Foreground The trajectory curve segment lying between 3–10 m in front of each camera position is used to guide the process. If the curvature at the start and endpoint of the segment is less than 1/30 m^{-1} and 1/50 m^{-1}, respectively, the track ahead is assumed to be straight, in which case the Z coordinates of the curve segment are lowered by the average sensor height—e.g. 1.45 m—in order to be approximately projected on the track floor, q.v. figure 7.17c. Otherwise, the track

ahead is assumed to be on a bend, in which case only the start point is lowered, as the floor at the endpoint is expected to be occluded by the inner track wall, q.v. figure 7.18c. The superpixels at the lower middle 20% of the image width are also assigned to the foreground.

Background On the same premise, the superpixels at the upper middle 10% of the image width are considered to belong to the background. In addition, the superpixels which include a feature match, and all the superpixels above them, are also assigned to the background if the feature point's object-space distance from the track's alignment is greater than 5 m. In addition, the upper left or right corner superpixels—for left or right bends, respectively—including their neighbours, are added to the background seeds.

The method was initially tested on an image set comprising 1111 frames with a resolution of 1280×720 pixels. The frames were extracted at approximately 1.5 m intervals from a 19 min long video, which was recorded with a hand-held camera during a track walk on the Sochi Sliding Centre, q.v. table 2.1, and was provided by the English Institute of Sport. The camera was moving along the track at a variable roll angle and height from the floor in order to capture the video from the point of view of the athlete during a run. The images were relatively oriented in Photoscan without using reference information; for the camera trajectory curve fitting and the photogrammetric model datum definition see section 8.1.

For the particular track, which is fully covered almost along its entire length, very few background seeds came from the sparse point cloud, as the roofs and blinds were closer than the 5 m threshold to the sensor trajectory. The selected value was considered an upper bound, which took into account tracks with kreisel bends that tend to rise higher than the rest of the track segments and can be adjusted to each track. Based on visual inspection, it was estimated that approximately 10% of the foreground on each image was erroneously assigned to the background on 5% of the image set, whereas more than approximately 90% of the background was segmented successfully on 30% of the images; the rest 65% of the image set included parts of the roof or blinds in the foreground. The inclusion in the foreground of the wooden guard on the outside of the bends, which in practice acts as a sidewall during a sled run, was considered desirable—whether it is covered with ice or frost, or not—but not essential. The reasons why some images were inaccurately segmented appear to be local in these examples, as the errors primarily occur where the blinds are very close to the track and have the same colour or at the lower start of the track where part of the sidewall is wooden and not covered in ice, q.v. figures 7.20 and 7.21, respectively. On some occasions, the image corner seeds have an undesirable effect on the segmentation which may be insignificant or not, q.v. figure 7.19. In most cases the part that is masked out is still visible on other views so the



(a) An image of the starting straight.

(b) Saturation channel.

(c) Superpixels and foreground (green) / background (red) pixel seeds.

(d) Segmented image of the starting straight.

Figure 7.17: Track segmentation on an image of a straight segment.

reconstruction may not be severely affected.

A 3D mesh of the Sochi track was generated in PhotoScan (2018), with and without using masks for the generation of the dense point cloud. The total gain in duration for the former was approximately equal to the time it took to segment the images. The effects of using the masks that were generated with this method on dense matching of higher resolution images and the surface reconstruction are investigated further in section 11.2.2. In any case, the reconstructed track surface will have to be segmented based on its geometric properties, q.v. section 8.3, as it is not guaranteed that its boundaries will be accurately segmented, q.v. figure 7.20a. Nevertheless, the segmentation process may reduce the geometric information that is generated on the surrounding environment and facilitates the start line detection process that is described in section 7.3.

7.3 Track Start Line Detection

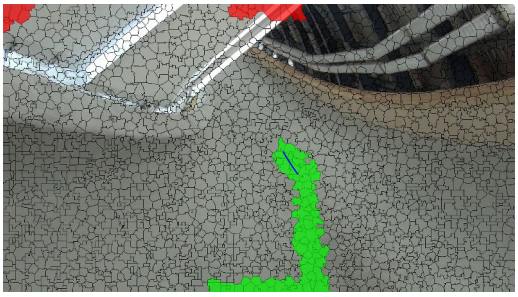
Defining the position of the track's start line endpoints within its 3D model, or along its alignment, is essential for the simulation of a sled run. As the process starts with the athlete in/on the sled-platform and the part where the sled is actually pushed is not simulated, an initial velocity is assigned to the sled at some point after the start line—where gravity takes over—which may vary



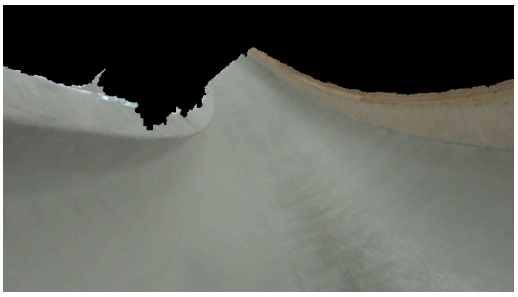
(a) Image of a straight segment.



(b) Saturation channel.

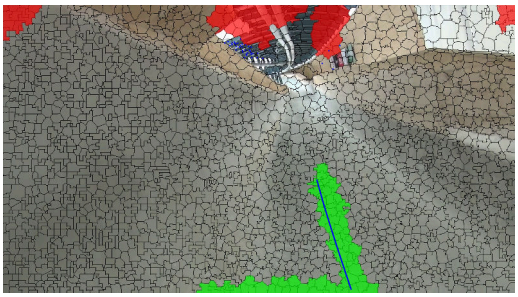


(c) Superpixels and foreground (green) / background (red) pixel seeds.



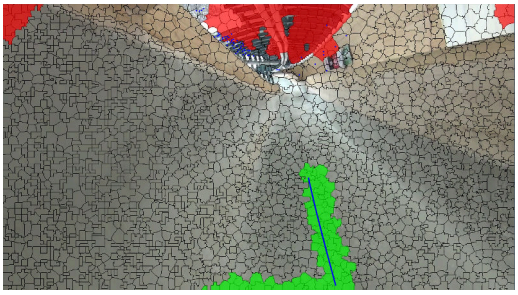
(d) Segmented image of the straight segment.

Figure 7.18: Track segmentation on an image of a bend.





(a) Superpixels, foreground (green) and background (red) pixel seeds, and segmentation of image 404.





(b) Superpixels, foreground (green) and background (red) pixel seeds, and segmentation of image 405.

Figure 7.19: Track segmentation on images 404 and 405 of the Sochi track, variously affected by misplaced background pixel seeds.

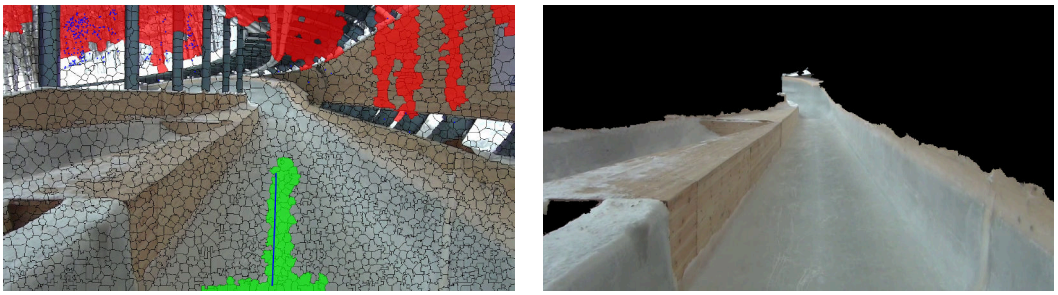


(a) Superpixels, foreground (green) and background (red) pixel seeds, and segmentation of image 590.

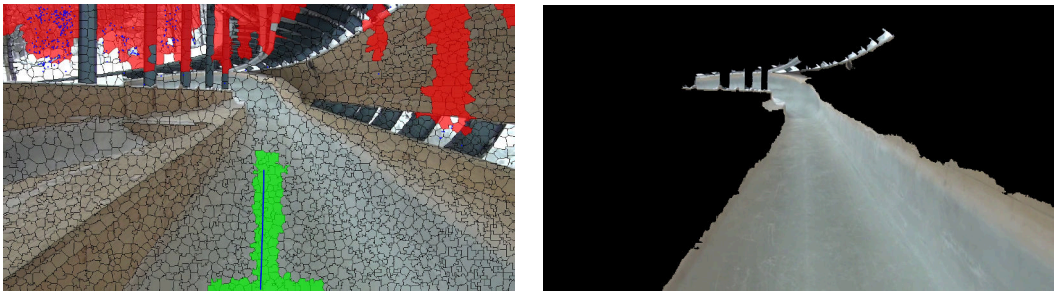


(b) Superpixels, foreground (green) and background (red) pixel seeds, and segmentation of image 591.

Figure 7.20: Track segmentation on images 590 and 591 of the Sochi track, variously affected by correctly placed background pixel seeds.



(a) Superpixels, foreground (green) and background (red) pixel seeds, and segmentation of image 425.



(b) Superpixels, foreground (green) and background (red) pixel seeds, and segmentation of image 426.

Figure 7.21: Track segmentation on images 425 and 426 of a part of the Sochi track where not both sidewalls are covered with ice.

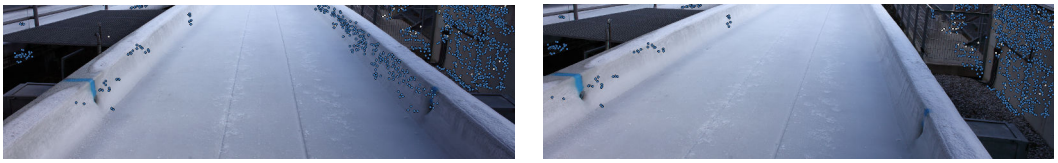
for each athlete, q.v. figure 2.1. The start line is not only essential for the simulation task but it can also facilitate the inference process due to the fact that the track's geometry up to that point can be considered to be relatively planar and horizontal, q.v. section 2.1; a transformation optimisation framework that uses this information for the datum definition is described in section 8.1. In theory, the start line coincides with the end of the push-off stretch where the downhill stretch begins; thus, it could be extracted based on its purely geometric attributes from the dense point cloud or 3D mesh. However, it may not always be possible to successfully reconstruct that part of the track, due to the reflective characteristics of the ice—see for example figure 5.7b with the result of the Winterberg trial where a significant of the push-off stretch is missing. A simple way to automate the start line detection on images of the track would be to require that the principal ray of, for example, the fifth and sixth stereo pair in the image sequence to be pointed at the left and right edge of the start line, respectively. As such a recipe may cause confusion in practice, a scenario was investigated where no such assumptions were made.

The start line endpoints can be described as two round black holes on the track's sidewalls, which host the sensors of the electronic timing system, q.v. figure 2.3. Nevertheless, they are not the same in every track and, unless it is known beforehand on which part of which image to look for them, their detection is not a straightforward task; even if a library of templates from the existing tracks was created, it may easily fail to generalise to recently modified or newly created tracks. The oblique imaging angles mean that they are barely visible; as they are imaged at various distances, only loose assumptions can be made on their image projection size and shape, which is not necessarily circular. In addition, there are many more structures in the vicinity that look like them; notice in figure 7.22 several dark blobs on white background that could be misinterpreted and how no feature points were matched or even detected on images of the Sochi, Igls, and Lillehammer track during the orientation process in PhotoScan (2018). Therefore, template matching, circle detection, or feature descriptors were not considered.

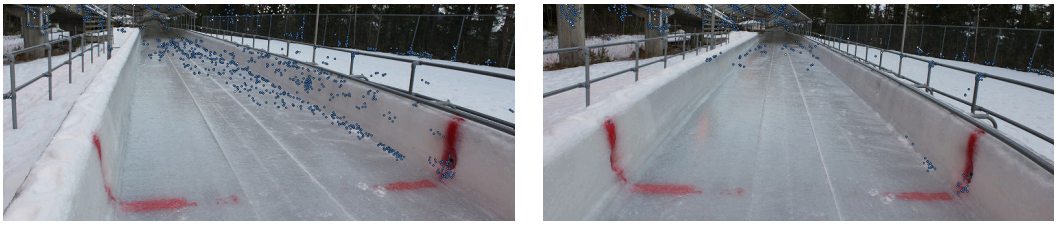
The Detection Process The input to this automatic process is the value channel of the downsized HSV image, q.v. figure 7.23b, which was used in the track segmentation process, q.v. section 7.2, and the detected image mask of the segmented track. Assuming that the first image was captured approximately 4–7 m behind the push-off line, only those images which were acquired between 8–23 m from the first camera station, are considered—q.v. yellow marks in figure 7.24d. The input image is converted to 9 binary images by the adaptive thresholding method described in Bradley et al. (2007), q.v. section 3.5.1, using 9 sensitivity values $adThS \in \{0.01, 0.05, \cdots, 0.29, 0.33\}$ and considering the dark pixels to belong to the foreground, q.v. figure 7.23c.



(a) No feature points were detected (○) and matched (●) around the endpoints of the Sochi track start line.



(b) Feature points detected (○) and matched (●) around the endpoints of the Igls track start line.



(c) Feature points detected (○) and matched (●) around the endpoints of the Lillehammer track start line.

Figure 7.22: The start line at the Sochi, Igls, and Lillehammer track. The start line endpoints' appearance on track walk images is distinctive but not necessarily round, unique or uniform.

All the dark pixels within the previously detected mask are potential projections of the start line endpoints. The candidate set of the start line endpoints' projections on the original images comprise the centroids of each connected component on the binary images. For each candidate point, at most one conjugate is selected from the points on the other images based on the epipolar geometry of the corresponding camera pair and the cross correlation of $w_b \times w_b$ image patches around the points, q.v. figure 7.24. In order to avoid using hard-coded values, the patch size w_b is defined in pixels with respect to the cameras principal distance, as the projection of a 0.25 m ground length parallel to the image plane at 5 m distance from the camera. From the points pairs with an epipolar constraint less than 0.15, the one with the maximum correlation score is selected, q.v. figure 7.24a–7.24c. Both points should be on the same side of the image with respect to the sensor path projection, in order for a match to be accepted—q.v. green, blue and red markers in figure 7.24d. Considering only the triangulated points that lie below the sensor path, within 15–25 m meters from its start, and at a lateral distance between 0.5–2.5 m, a small subset remains scattered around the start line, q.v. blue markers in figure 7.24d. One option would be to project them back on the images and refine the detection process. Instead, the problem is solved in a brute-force combinatorial fashion in the sense that from all the pairs between the points on either side of

the sensor path, the one that minimises the sum of squares of the distance error with respect to the track's nominal start line width, q.v. section 2.1, and distance along the sensor path, is selected.

The proposed algorithm was tested on an image sequence that was collected along the Lillehammer track's push-off stretch, q.v. red markers in figure 7.24d. The total and horizontal RMSE of the endpoint positions, with respect to manual observations on the same images, was 19 cm and 11 cm, respectively. The error in the estimation of the start point position along the track alignment, after averaging the left and right endpoint projections, was 1 cm.

Chapter Summary

Three small-scale computer simulation experiments were conducted with respect to assessing the capabilities of simple imaging configurations—which can be easily implemented with a hand-held camera, without additional equipment—for 3D reconstruction. The simulation process took into account the non-deterministic nature of the geometric distribution of the points within a non-signalised scene that were used in the bundle adjustments. As expected, linear camera motion, i.e. a sensor trajectory parallel to the track's alignment, is the least reliable; image resolution is important, while enlarging the field of view with a wide lens does not necessarily improve the result. On the one hand, there was no indication that increasing the complexity of the imaging configuration would improve the reconstruction results. On the other hand, none of the configurations was superior for both sparse and dense reconstruction and further investigation is required, preferably within full-scale trials.

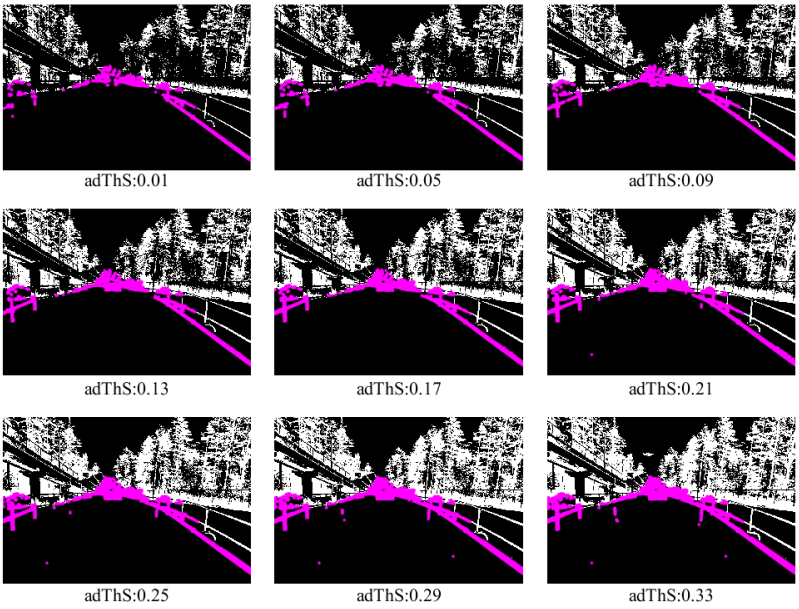
A track segmentation process was also described, which automated the interactive graph-based "Lazy Snapping" image segmentation algorithm by utilising 3D information from the track's sparse reconstruction. It was demonstrated that it can successfully segment the track but further investigation is required with respect to the effect of incorrectly segmented images on the surface reconstruction result. In any case, it can facilitate the brute-force start line detection process by reducing the number of image projections that are considered as potential endpoints.



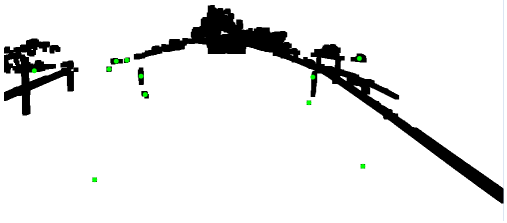
(a) Input image.



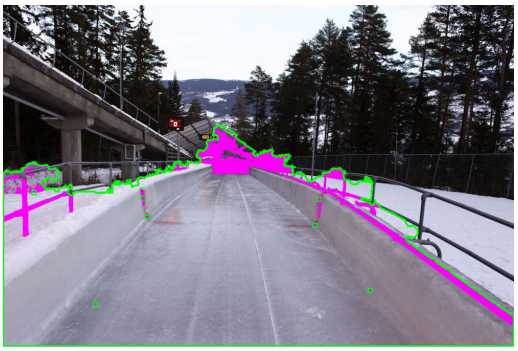
(b) The Value channel after the RGB to HSV transformation.



(c) Adaptive thresholding at nine sensitivity levels.

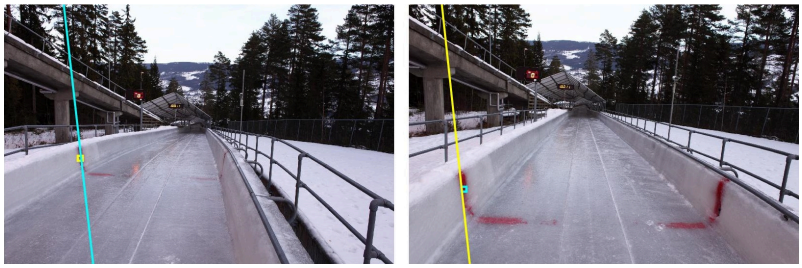


(d) The combined result of adaptive thresholding. The centroid of each connected component on the binary images is indicated by a green circle.

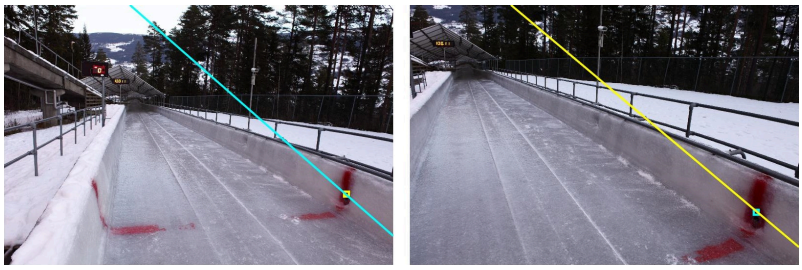


(e) The start line image projection candidates within the detected track surface mask.

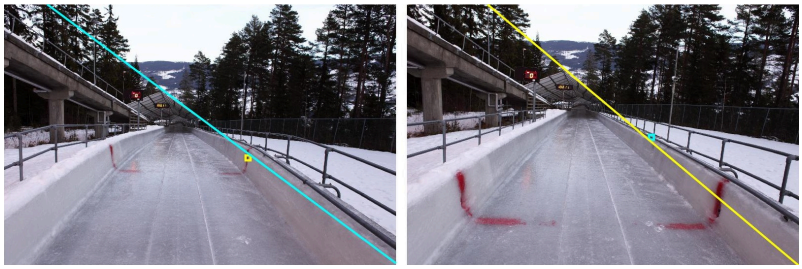
Figure 7.23: The automatic start line endpoint detection process on images of the Lillehammer track.



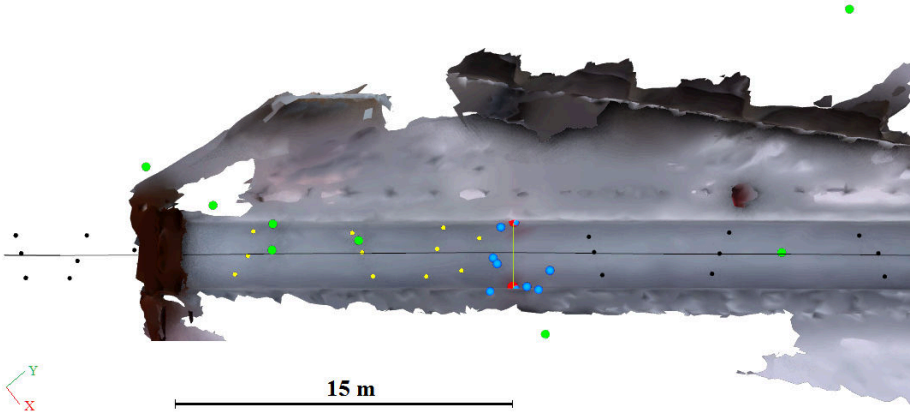
(a) Correctly matched points on the left endpoint.



(b) Correctly matched points on the right endpoint.



(c) An incorrectly matched pair.



(d) The detected start line endpoint candidates and camera positions overlaid on a 3D mesh (top view).

Figure 7.24: Matching and triangulating the image projections of endpoint candidates for the detection of the Lillehammer track start line.

(—) start line, (●) detected endpoints, (●) filtered candidates, (●) candidate endpoints, (●) camera positions, (●) cameras used in endpoint detection, (—) sensor path, (■)(■) candidate endpoint image projection and (—)(—) corresponding epipolar line.

Chapter 8

Geometric Scene Reconstruction

In the previous chapter, planning and preprocessing methods for image-based 3D scanning and reconstruction were investigated. This chapter assembles geometry processing tools for the extraction of the parametric 3D model of a track's surface from a 3D mesh that contains noise, gaps, and clutter. The proposed techniques were designed to deal with the challenges that stem from image-based scanning but can be applied to the output of a laser scanning survey, as well. More specifically, a method for the automatic extraction of characteristic cross-sectional curves along the track's alignment is described. These curves should only represent the track's surface and reflect the design intent of the typical cross sections that were used for the track design. The result is the track's representation as a generalised cylinder, which can be used with a standard lofting or path-based skinned surface approximation algorithm, q.v. section 4.3. Section 8.2 is concerned with curve fitting as part of a horizontal alignment extraction process; in particular, fitting a curve model that is extensively used in road and track design, namely clothoid splines, is considered. Section 8.3 describes a path-based sweeping-and-morphing algorithm that uses the extracted characteristic cross sectional curves to fill in missing information along the track. The initialisation of the track's alignment and the datum definition of a photogrammetric model in the absence of adequate reference measurements are addressed in section 8.1.

8.1 Track Alignment and Datum Definition

The optimal way to compensate for the non-linear distortions of the photogrammetric model of an elongated object is by measuring ground control points (GCP) along its length and using them to constrain the bundle adjustment solution, thus, solving the datum definition problem, as well, q.v. section 3.3.1. The GCP measurements are commonly collected by on-site GNSS surveys or

from high resolution satellite images and dense terrain or surface models. Both options require site reconnaissance in order to identify suitable points which can also be measured by a hand-held sensor from within the track. In theory, this is only required once per track but their visibility on images taken from inside the track can be compromised by roofs, blinds, or even snow. On the one hand, the additional overhead of carrying a geodetic grade GNSS receiver on site is incurred, in order to achieve the required GCP positioning accuracy within a short time frame. On the other hand, even though an orthoimage of the track may be available freely and publicly, a remotely sensed elevation model can only provide approximate information about the geometry of a track whose entire length is roofed, as the track's surface itself cannot bee seen.



(a) Winterberg track. **(b)** Sochi track.

Figure 8.1: Observing points, which belong to the same cross section of the track, in an image in order to infer the scale of a photogrammetric model based on the nominal track width along the straight segments, may be challenging even for a human operator.

In this section, an optimisation framework for the datum definition of the 3D model of a track when no reference measurements are collected during the image-based scanning is considered. Assuming that the non-linear distortion component of the photogrammetric model is of low spatial frequency and that a coherent relative scale for the model can be estimated by using a calibrated camera with stable geometry, if the scale and orientation of the photogrammetric model $XYZ^{M'}$ with respect to a local coordinate system XYZ^{W}—parallel to the world's reference frame with the negative Z-axis along the earth's gravity direction at the origin, q.v. figure 3.5—could be approximated, the track's 3D model could be useful for training purposes, given that its safety can be verified by a subsequent trajectory simulation, even if it is not be an exact representation of the scanned track. The hypothesis was that inferring the scale factor and rotation of the track model based on the redundancy from all the available prior knowledge of the track's geometry can be a reliable solution. The idea was that the similarity transformation between the photogrammetric

model and the final 3D model of the track could be continuously refined as more information becomes available during the 3D reconstruction, so that each approximate estimation could facilitate the inference process at the next step.

8.1.1 Initialising the Track's 3D Alignment

In the first instance, an approximate initialisation of the scale and orientation of the photogrammetric model is required, so that certain information about the track, i.e. its nominal width along the straights and the fact that the camera stations are "above" the track's floor, can be used for inference tasks—such as the start line detection, q.v. section 7.3—and towards extracting its 3D model, q.v. section 8.3. As discussed in section 3.2.2, using the rotation angles and length estimations from an array of accelerometers, gyroscopes, and magnetometers attached to the camera requires a tedious calibration process and may yield unreliable results. Scaling and rotating a 3D model, so that it assumes the correct size and orientation, can be manually accomplished by identifying objects of known dimensions that can be presumed to be vertical or horizontal and using them as reference. However, within an automated workflow, user interaction should only be required during the data acquisition process which, together with the nature of the collected data in combination with prior information, should be able to accommodate a seamless 3D reconstruction process.

Automating the aforementioned approach requires algorithms for the detection of objects with such properties in a 3D point cloud or mesh, e.g. light poles or building edges, whose existence, however, is not guaranteed. The approach that was used in the preliminary trials, q.v. chapter 5 for the scale definition could also be followed; nevertheless, even if the user measured the track widths along the straight segments—instead of assuming the nominal width for the track at the iced state, q.v. section 2.1—making the corresponding observations on the images can be quite challenging even for a human operator, let alone an algorithm; q.v. figure 5.4b cf. figure 8.1. In addition, the transformation may be required prior to the reconstruction in order to guide the image segmentation process, q.v. section 7.2.

It is assumed that a piecewise linear approximation of the track's alignment is available. If the track was scanned using a hand-held laser scanner or camera in an orderly fashion—e.g. from start to finish—then the sensor's trajectory can be used to initialise the alignment. In the general case, where no such initial alignment exists or if the image list is not ordered, the initial solution can be retrieved by a skeleton (Cornea et al., 2007; Tagliasacchi et al., 2009; Bucksch, 2011), medial axis (Blum, 1967; Held, 2001; Dey, 2006), or principal curve (Hastie et al., 1989; Delicado, 1998; Kegl et al., 2000) extraction algorithm. Given that the exterior orientation of n images which were

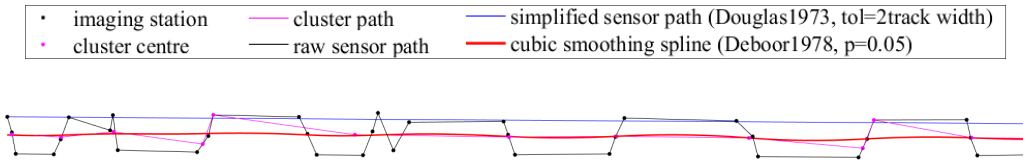
captured sequentially along the track is known up to a scale, the first step is to initialise the track's alignment as a polygonal line that represents the piece-wise linear, along-track sensor trajectory from the ordered coordinate list with the image locations, q.v. figure 8.2. The polyline's length is used in order to estimate an initial scale factor sc_{init} for the photogrammetric model based on the assumption that the sensor's trajectory length is between the nominal competition and total lengths, q.v. table 2.1, or equal to a user-defined estimation of the scan length. Note, however, that if a generic block other than the first one in figure 7.2 was used for the image acquisition, a length estimation for the scanned track alignment based on the sensor path would be inaccurate. In order to keep this step agnostic to the geometry of the implemented generic block, the scale estimation is based on the $n/2$ cluster centres estimated with the k-means algorithm (Arthur et al., 2007). Considering the average track lengths and an estimation error up to 100 m, it can be assumed that for any track in general, the length estimation error will be below 8%. Given the value for sc_{init}, an approximate chord-length parameterisation is estimated for the n image locations by simplifying the polygonal line which sequentially connects them using the Douglas-Peucker (Douglas et al., 1973) with a simplification tolerance equal to the nominal track width along the straight segments, i.e. $tol_{DP} = 1.4$ m; for the first 75 m along the starting straight it is set to $tol_{DP} = 3$ m. Finally, the track's alignment is initialised in a continuous form by fitting a cubic smoothing spline (De Boor et al., 1978; Matlab, 2017a) to the parameterised discrete list of the camera's stations along the track; a smoothing parameter equal to 0.05 worked well for the simplified data sets that were considered in this study.

8.1.2 Optimising the Datum Definition

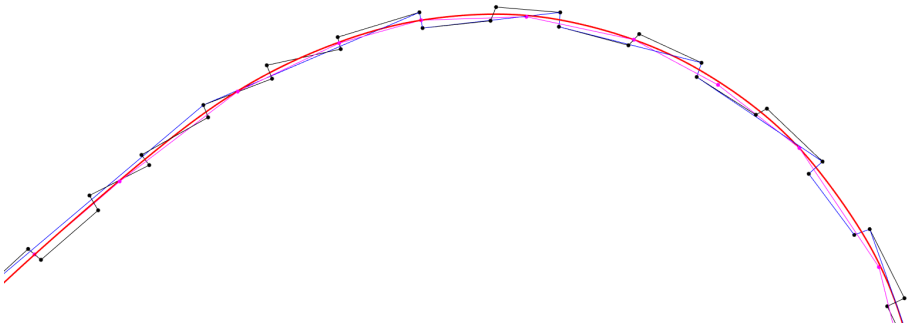
The datum definition of the track's photogrammetric model is initialised by a transformation that rotates the camera coordinate system of the first image so that it becomes parallel to a local coordinate system, based on the known sensor coordinate system convention and assuming that it was captured in landscape mode—e.g. the positive Y^M axis can be defined as being parallel to the positive Z^W axis and the positive X^W and X^M axes parallel and horizontal, q.v. equation 8.1 and figure 3.5.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} XYZ^{M'} = \boldsymbol{X}YZ^M \approx \boldsymbol{X}YZ^W \tag{8.1}$$

Note, however, that the photogrammetric model may have been defined as being parallel to the camera coordinate system of any one of the images, i.e. one of the images that belongs to the stereo



(a) Track alignment initialisation along the starting straight (top view).



(b) Track alignment initialisation along a bend segment (top view).

Figure 8.2: Track alignment initialisation by fitting a cubic smoothing spline to the piece-wise linear sensor path.

pair that was oriented first. As this transformation can be far from the correct one, a similarity transformation optimisation scheme is proposed.

The geometric attributes of the track that are used, as they became available, in order to infer an optimal transformation are the following, q.v. section 2.1 and IBSF (2017):

- the length and gradient of the push-off stretch are approximately 15 m and -2%, respectively.
- the vertical drop between the highest and lowest track floor elevations is approximately known
- the track's alignment is downhill for the most part of its length, with the maximum gradient of approximately 20% occurring along the deceleration stretch towards the finish line; other parts may have positive gradient, as well, but the average gradient is negative and approximately known.
- the cross section of the track's floor is approximately horizontal
- the width of the track's cross sections along the straight segments is approximately known; its nominal value is approximately equal to 1.4 m and 1.5 m with or without the ice layer, respectively.

The target values for these attributes may be inaccurate or inadequate to uniquely estimate the transformation and will only approximately correspond to those of the sensor's path, which cannot

be expected to be exactly parallel to the track's as-designed centreline alignment. Therefore, a multi-objective genetic algorithm (Deb, 2001; Matlab, 2017e) is proposed for the optimisation of the photogrammetric models' transformation, q.v. section 4.1. The similarity transformation of the model, whose centroid is assumed to coincide with the coordinate system's origin, is parameterised with a scale factor sc and an axis-angle representation for the rotation, where the rotation angle θ is encoded as the norm of the rotation vector $\boldsymbol{r} = [x_r, y_r, z_r]^T$ as $\theta = \sqrt{x_r^2 + y_r^2 + z_r^2}$. These are the parameters that will be optimised according to $\underset{\boldsymbol{r},sc}{argmin}(C_{ci})$, q.v. equations 8.2–8.7. From the set of the Pareto-optimal solutions, the one with the minimum sum of squares of the normalised cost functions $\sum \bar{C}^2$ is selected as the final solution.

$$C_1^{sim} = (\frac{100}{st}\sum_{k=1}^{q-1}\frac{Z_{q+1} - Z_q}{L_q^H} - G^{push})^2, \text{ with } q \leq st \tag{8.2}$$

$$C_2^{sim} = (100\frac{Z_b - Z_1}{\sum_{k=1}^{b-1} L_k^h} - G^{avg})^2 \tag{8.3}$$

$$C_3^{sim} = ((maxZ_q - minZ_q) - VDrop)^2 \tag{8.4}$$

$$C_4^{sim} = \sum_{G_k<0, k<b} L_k^S (\sum_{k=1}^{p-1} L_k^S)^{-1} \tag{8.5}$$

$$C_5^{sim} = (ww_q - w^{track})^2 \tag{8.6}$$

$$C_6^{sim} = (100 G_q^{floor})^2 \tag{8.7}$$

such that

$$0 \leq theta < 2\pi \tag{8.8}$$

$$0.8 \leq sc \geq 1.25 \tag{8.9}$$

$$maxG_k \leq 30\% \tag{8.10}$$

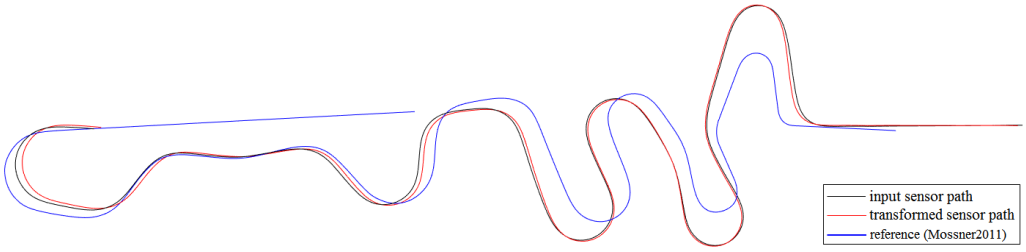
$$minG_k \geq -40\% \tag{8.11}$$

$$\sum_{k=1}^{b-1} L_k^S > 0.5 \sum_{k=1}^{p-1} L_k^S \tag{8.12}$$

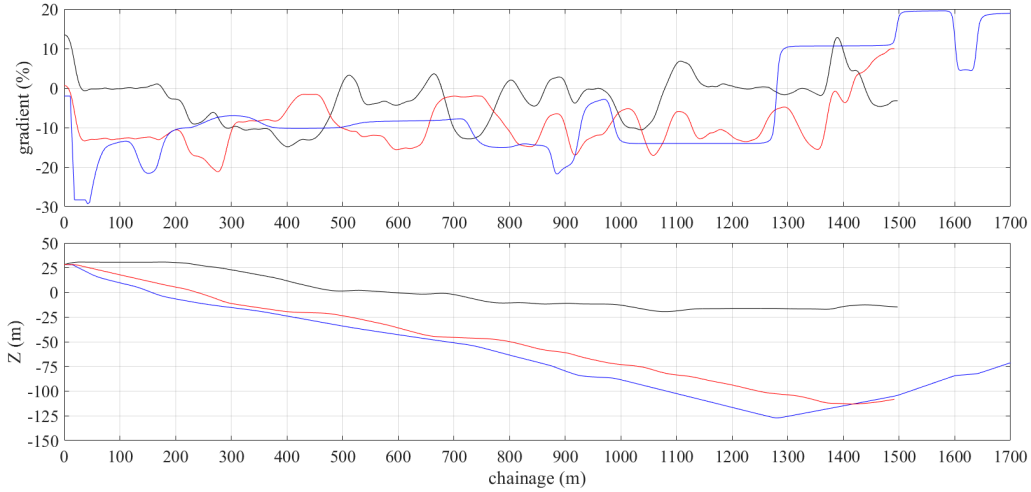
where

C_1^{sim}	=	starting straight gradient cost	C_2^{sim}	=	average gradient cost
C_3^{sim}	=	vertical drop cost	C_4^{sim}	=	uphill cost
C_5^{sim}	=	section width cost	C_6^{sim}	=	across-track floor gradient cost
m	=	the total number of points that were sampled along the alignment for its discretisation			
k	=	curve segment index; $k \in \{1, q-1\}$	q	=	points index $q \in \{1, m\}$
st	=	index of the start line	b	=	index of the lowest point, $Z_{q,q\neq b} > Z_b$
w^{track}	=	nominal track width	G^{push}	=	nominal starting straight gradient
G^{avg}	=	nominal average gradient	$VDrop$	=	nominal maximum vertical drop
L_k^s	=	slope length of k_{th} segment	L_k^h	=	horizontal length of k_{th} segment
G_q^{floor}	=	the across-track floor gradient at each point q, estimated from the track's cross sections			
ww_q	=	the wall-to-wall track width at each point q, estimated from the track's cross sections			

$Z = \boldsymbol{f}_1(Z_{sc}^M(\boldsymbol{r}, sc))$, $L = \boldsymbol{f}_2(\boldsymbol{SC}^M(\boldsymbol{r}, sc))$, $G = \boldsymbol{f}_3(\boldsymbol{SC}^M(\boldsymbol{r}, sc))$, $F = \boldsymbol{f}_4(\boldsymbol{CS}^M(\boldsymbol{r}, sc))$, and $ww = \boldsymbol{f}_5(\boldsymbol{CS}^M(\boldsymbol{r}, sc))$ are all functions of the alignment's $\boldsymbol{SC^M} = [X_{sc}^M, Y_{sc}^M, Z_{sc}^M]$ and the ex-



(a) The horizontal projection of the sensor path before/after the transformation.



(b) The elevation and gradient along the sensor path before/after the transformation.

Figure 8.3: Approximate transformation of the photogrammetric model of the Whistler track compared to a reference track alignment extracted from Mossner et al. (2011).

tracted cross-sectional curves' coordinates $\boldsymbol{CS^M} = [X^M_{cs}, Y^M_{cs}, Z^M_{cs}]$ in the reference system of the photogrammetric model and the transformation parameters $\boldsymbol{r}$ and sc.

The first three properties also apply, to a certain extent, to the sensor's trajectory acting as a proxy for the track's alignment; thus, they can be used directly after the image orientation process. The nominal maximum vertical drop can be retrieved from the official track description or from measurements either on site—e.g. using a barometric altimeter—or on a digital terrain model of the site. Unless the value for the vertical drop is considered to be reliable, the scale factor sc is not used in the optimisation; instead, it is initially fixed according to an estimation of the total sensor trajectory length, q.v. section 8.1.1. The last two items on the list refer to the track's cross sections and can only be taken into account after their extraction, q.v. section 8.3.2, but they can be integrated in the same optimisation scheme. If no additional information that can define the scale is available, i.e. start or push-off line widths, it is fixed according to the mean width of the track's cross sections along the straight segments and only equation 8.7 is used in a standard gradient descent-based least squares optimisation process.

This process was tested on the dataset from the Whistler preliminary trial that was presented in section 5.1. Firstly, a cubic spline was fitted to the camera trajectory, q.v. section 8.1.1, and the scale was defined by assuming that the length of the scanned track was $\hat{L}_{total}^{track} = 1500$ m. At this initial stage, only the rotation is optimised using equations 8.2–8.5 as the cost functions and the nominal reference values for the gradient and the maximum vertical drop, q.v. section 2.1 and table 2.1. Figure 8.3a shows the horizontal projection of the sensor path before and after the optimisation; for the purpose of illustration, both curves were registered to a reference track alignment for the Whistler track which was extracted from Mossner et al. (2011). The apparent difference between the two paths and the reference curve is due to a combination of the scale and rotation estimation errors. Nevertheless, figure 8.3b shows that the improvement to the along-track vertical profile can sufficiently support the assumption that the Z coordinate of the track's floor at each camera station is less than the Z coordinate of the corresponding image position. In other words, an approximate vertical profile of the track's floor can be acquired by projecting the sensor path on the mesh along the negative Z axis, in order to guide the image segmentation and cross section extraction process, q.v. sections 7.2 and 8.3.1, respectively.

8.2 Horizontal Alignment Extraction

A three-stage curve fitting algorithm for the extraction of the track's horizontal alignment is proposed in this section; it comprises a curve segmentation followed by a control polygon extraction and clothoid spline fitting process, q.v. figure 8.4, based on a decompose and recombine strategy (Polya, 1945) which gradually refines a global solution that is initialised locally in curvature and object space. The input is the 2D coordinates $[X_q, Y_q]$ of a set of m points p_q, with $q \in \{1, 2, \cdots, m\}$, which have been measured along the centreline of a track or a road. The values of two user-defined parameters are required: an estimation for the measurement noise standard deviation σ_n and a distance D^{cloth}, which defines the radius of the neighbourhood that is used for the discrete curvature estimation at each point—essentially a smoothing factor—and as an indication of the total length of the shortest control polygon edge and bend segment. It is assumed that an approximate arc-length parameterisation t is available for the point set—if not, it can be estimated using the approach that was described in section 8.1.1 with $tol_{DP} = 3\sigma_n$. It is also assumed that a tangent—i.e. a straight line segment—always exists at the beginning and end of the alignment. If this is not the case, the input can be extended at the endpoints or the first and last D^{cloth} metres will be defined as tangents in order to construct the control polygon. In addition, it is assumed that a tangent always exists between two arc segments and that a transition clothoid segments either

exist on both sides of an arc or on neither one. In other words, each bend is modelled using a clothoid-arc-clothoid sequence, called a combined bend, q.v. figure 8.7b, or a simple circular arc. The lengths of the tangent and clothoid segments can be reduced to zero during the optimisation step, q.v. section 8.2.3.

8.2.1 Curvature-Based Alignment Segmentation

The output of this stage is the alignment's initial primitive type sequence and an approximation of the length and curvature for each segment, together with the corresponding input points for each one. The curve segmentation stage begins with the estimation of the signed discrete curvature $\kappa(t)$ at a subset of the input which includes points whose distance from the previous and next points along the curve is greater than $10\sigma_n$ ($\kappa_{q'}$ with $q' \subset q$, such that $t_{q'} - t_{q'-1} > 10\sigma_n$ and $t_{q'+1} - t_{q'} > 10\sigma_n$) in order to prevent the curvature estimation from few noisy points located closely together, e.g. the three points at the middle of the circular arc in figure 8.7b. For the estimation of κ at each point $p_{q'}$, the algebraic circle fitting method (Taubin, 1991; Chernov, 2010, 2012) is used—as a good compromise between the geometric and osculating circle fitting methods in terms of the robustness and computational effort—based on all neighbours $p_{q''}$, such that $q'' \subset q'$, $t_{q'} - D^{cloth} < t_{q''} < t_{q'} + D^{cloth}$. If no points exist within D^{cloth}, the calculation is based on the two adjacent points $p_{q'-1}$ and $p_{q'+1}$. The curvature sign convention is that left turning bends have positive curvature.

The alignment's segmentation begins with the detection of zero-crossings t^o, where $\kappa(t^o) = 0$, on the approximate curvature profile of the underlying true alignment, q.v. figure 8.5 top. This approach may fail to segment compound bends or adjacent bends with the same curvature sign that are connected by short tangents, as the latter may be smoothed out and prevent the curvature

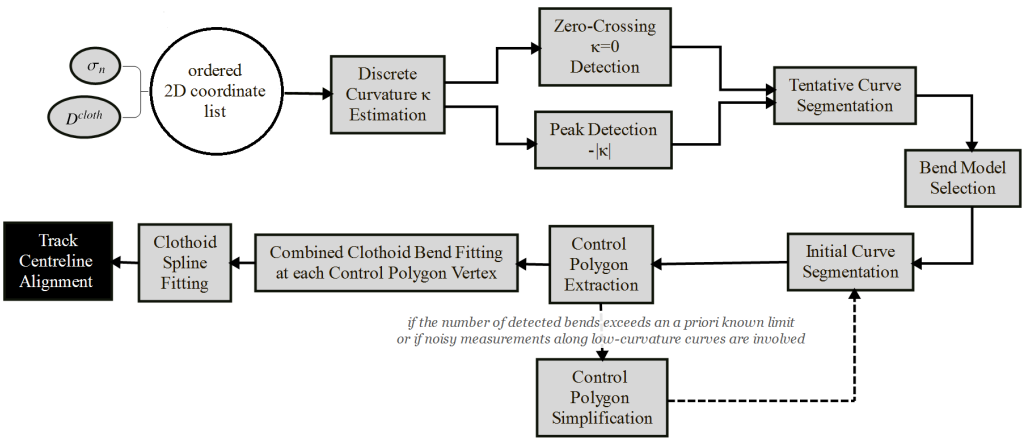


Figure 8.4: Clothoid spline fitting workflow.

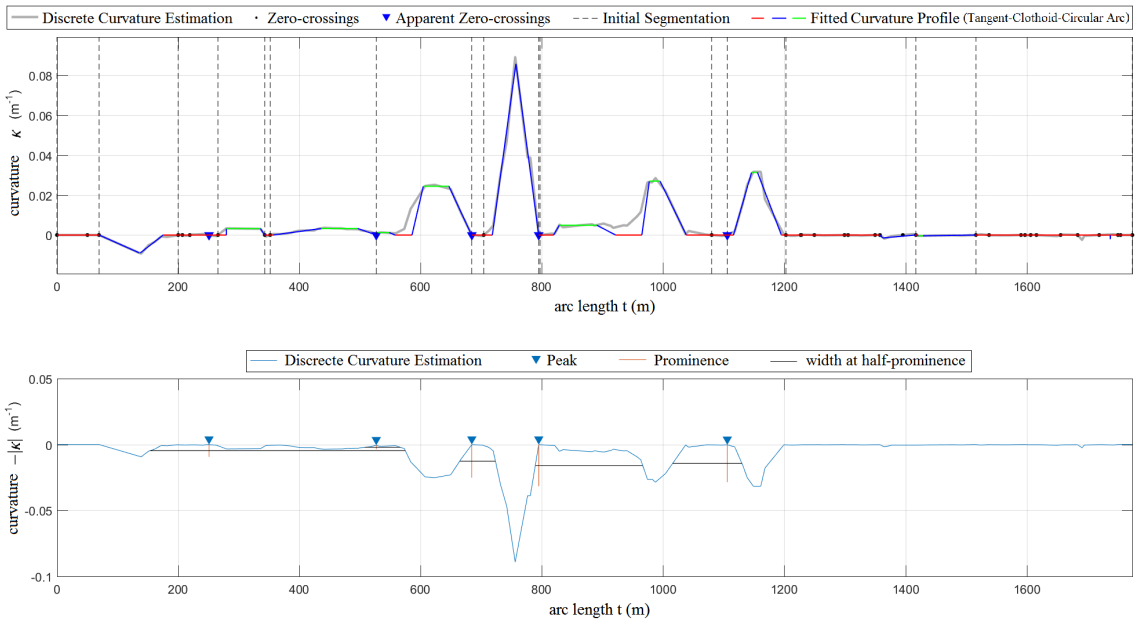


Figure 8.5: Curvature-based alignment segmentation (Horizontal alignment extraction - 2nd Experiment: the result of the proposed method with $D^{cloth} = 30$ m and $\sigma_n = 0.33$ m).

from reaching zero. As a result, two or more bends could be merged together and treated as one—which means that a single radius would be fitted to two arcs with different radii. To compensate for this undesirable effect, in addition to the zero-crossings, local maxima (Matlab, 2017c) $t^{o'}$ of the function $-|\kappa|$ are detected, q.v. figure 8.5 bottom. As peaks occur not only at the points of interest but also along tangents and arcs, in order to prevent circular arcs from getting over-segmented because of spikes due to noise, only prominent peaks with a relative height with respect to the highest of the adjacent minima greater than a predefined value are taken into account—a minimum prominence value of 0.003 m^{-1} was used in this study. On the one hand, this curvature threshold will potentially also pick up very small discrepancies in radius estimation along arcs with small radii, e.g. only about 0.29 m for arc radius of 10 m, which can lead to curve over-segmentation. However, arcs with a small radius tend to have smaller lengths so they are more efficiently smoothed. Thus, increasing the threshold is not necessary as it would mean missing some of the actual zero-crossings. On the other hand, if the maximum radius of an erroneously merged bend is larger than 333.33 m, it will not be separated at this step. Nevertheless, if the remaining unsegmented bends include no more than 2 arcs, they can be recovered at the next stage, q.v. section 8.2.1.1—in fact, more than 2 arcs could be recovered but this would significantly increase the computational effort.

The candidate zero-crossings list t_c^0 comprises the combined set of zero-crossings $t_c^0 = t^o \cup t^{o'}$ and includes the final number of z zero-crossings t^0 which will define the segmenta-

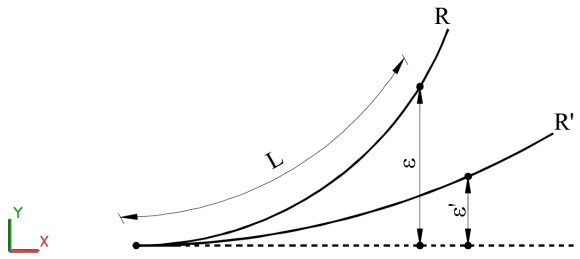


Figure 8.6: Circular arc segment deflection ε from the tangent with respect to arc length L and radius R.

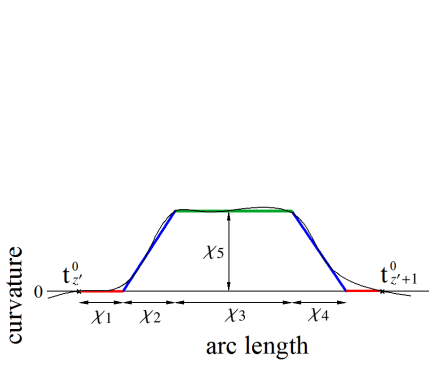
tion to $z-1$ segments. Any extraneous zero-crossings—due to noise—along over-segmented tangents are purged by the following scheme: The final t^0 list is created by iterating over all t_c^0, starting at the beginning of the alignment ($t_1^0 = 0$). A candidate t_c^0 is added to the list only if the length of the created segment $\Lambda_c = t_c^0 - t_{c-1}^0$ is greater than $0.25D^{cloth}$ and its estimated deflection ε_c, q.v. figure 8.6, is significant with respect to the expected noise amplitude, i.e. $\varepsilon \geq 3\sigma_n$, in which case the corresponding segment is labelled as a bend; it is otherwise considered to be a tangent. By using a deflection threshold, both the total turning angle as a result of the bend, expressed by the total integral $E_c = \int_{t_{c-1}}^{t_c} |\kappa(t)|dt$ of the curvature plot between the current and the examined zero-crossings, and the average curvature R_c^{-1} are taken into account according to $\varepsilon_c = R_c(1 - cos(E_c))$, where $R_c = \Lambda_c E_c^{-1}$. This strategy is used in order to prevent over-segmentation of tangents as insufficient smoothing along segments with lower curvature will create many erroneous zero-crossings.

Even if a hard-coded threshold had been used in order to label segments with low curvature as tangents—the approach that is followed in most studies, q.v. section 4.2.2—the presence of spikes due to noise requires an additional step in order to verify that these segments are indeed bends. In any case, dividing a long arc with low curvature into more arcs of equally low curvature is more desirable than ignoring it altogether and treating it as a tangent. The fact that this approach may yield successive control polygon edges that are near parallel, is addressed in section 8.2.2.

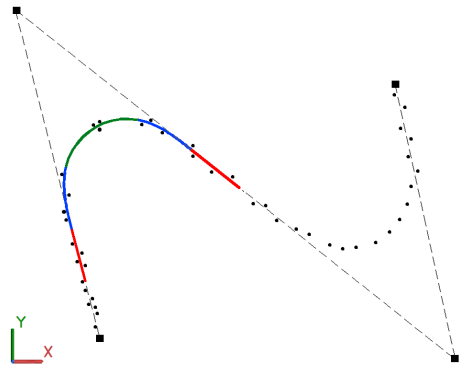
8.2.1.1 Bend Model Selection

In order to detect bends that the previous step failed to split—note for example in figure 8.5 than no zero-crossing was detected at $t \approx 580$ and $t \approx 930$—and acquire better initial values for clothoid spline fitting in object space, three piece-wise linear curvature models—comprising a primitive sequence combination of tangents, arcs, and clothoids, q.v. figure 8.7—are fitted in a least-squares sense in curvature space to each segment that has been identified as a bend. The simplest type is that of a combined bend, q.v. figure 8.7a. This includes simple bends without transitions

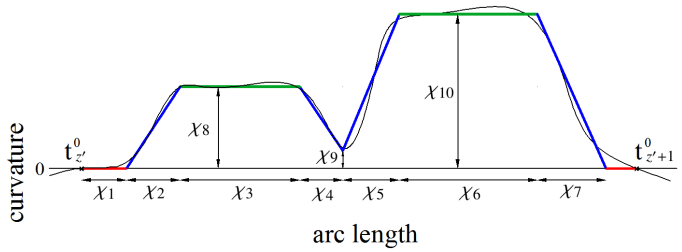
that appear as such due to smoothing. The second type, shown in figure 8.7d, is a clothoid-arc-clothoid-arc-clothoid sequence, representing two merged simple arcs having their common tangent smoothed out and appearing as a clothoid segment. Finally, a clothoid-arc-clothoid-clothoid-arc-clothoid sequence representing two merged combined bends without a tangent between them, see figure 8.7c. Each model is parametrised by the offset from the beginning of the corresponding segment ($t^0_{z'}$ where $z' \in \{1, 2, \cdots, z\}$), the primitives' lengths, and the arcs' curvature, q.v. figure 8.7a–8.7c. The value that is minimised is the sum of squares of the curvature difference at corresponding locations along the curve $\underset{\boldsymbol{\chi}}{argmin}(\kappa(t) - \kappa'(t))$ with $t^0_{z'} \leq t \leq t^0_{z'+1}$.



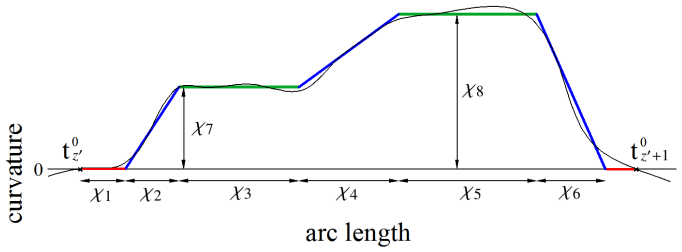
(a) A combined clothoid bend in curvature space.



(b) A combined clothoid bend in object space.



(c) Two combined clothoid bends.



(d) A compound circular bend.

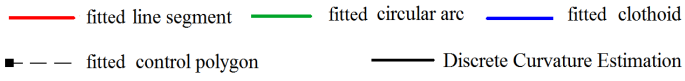


Figure 8.7: Fitting individual bend models in curvature and object space.

The last two models can provide a good, yet over-parameterised, fit for simpler bends, unless certain constraints are imposed in order to prevent that from happening. These constraints include

restrictions on the minimum length of each bend, which should be greater than D^{cloth}, and the ratio between the radii of the two bends—for the trials presented in section 8.2.4, the maximum radius should be twice the minimum between the two. The simplest possible model is assigned to a bend, unless the next most complicated one fulfils these constraints and improves the fitting error by more than a pre-specified percentage—in this study 50%. In a last attempt to prevent over-segmentation of bends in curvature space, if two consecutive arcs are closer than $0.5D^{cloth}$ and the ratio of the absolute difference of their radii to the minimum radius is less than 0.25, they are merged and treated as a single bend.

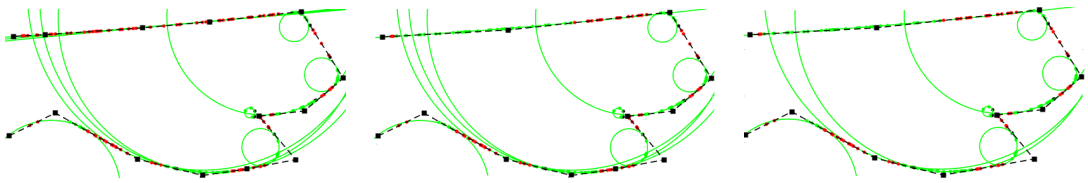
8.2.2 Control Polygon Extraction

The alignment's curvature profile may be good enough for the initial segmentation but not necessarily for estimating the total number of circular arc segments, their length, or the absolute value of their radii, as noise and over-smoothing may severely affect the estimation at both high and low curvature segments. A reasonable segment sequence, which adheres to certain design principles and is justified by the measured points, can be inferred by extending the model selection process, q.v. section 8.2.1.1, from curvature to object space, instead of trying to minimise the total number of segments based solely on their curvature profile.

To this end, the control polygon is initialised by intersecting consecutive line segments that have been fitted to the 2D coordinates of point subsets that have been identified on the curvature plot as belonging to tangents. If the subset includes less than 2 samples or corresponds to a curve segment shorter than D^{cloth}, it is expanded on both sides until it does. The MSAC (Torr et al., 2000) flavour of the RANSAC algorithm (Fischler et al., 1981) implemented in Zuliani (2012) is used to not only filter out outliers but also ignore the points that do not belong to the corresponding model type. The inlier selection threshold σ_R required by RANSAC is set equal to $\sigma_R = 3\sigma_n$. In addition, a circle is fitted in a least-squares fashion (Taubin, 1991; Chernov, 2010, 2012) to each point subset that lies along segments that have been identified as bends, in order to facilitate the control polygon simplification.

The initial control polygon is simplified, in order to remove potentially over-segmented bends, as follows:

- the Douglas et al. (1973) algorithm with a tolerance value euqal to $tol_{DP} = 3\sigma_n$
- vertices with a deflection angle lower than a predefined threshold are removed—$\Delta_{tau} < Delta^{min} = 1^{\circ}$



(a) Iteration 1. **(b)** Iteration 2. **(c)** Iteration 3.

Figure 8.8: A control polygon simplification example, q.v. figure 8.13a.

- vertices adjacent to successive circular arcs are merged if their centres are within a fraction of the radius of the largest one among them —0.1 max(R) used in the example presented in this article.
- if a control polygon edge is shorter than D^{cloth}, then one of the adjacent vertices, excluding the ones where the central angle a is greater than π, is removed, q.v. figure 8.9
- if the total length of a bend, which is estimated from the central angle and radius at each polygon vertex assuming a simple bend model, is less than D^{cloth} the vertex is removed.

The process is performed iteratively, q.v. figure 8.8, until no more vertices that fulfil the aforementioned criteria exist in the control polygon or if a predefined number of vertices is reached, e.g. the total number of track bends.

Calculating the deflection angle $\Delta_\tau \in (0, \pi)$ at each vertex between two control polygon segments is straightforward, q.v. figure 8.9. However, the estimation of the central angles, $\alpha_\tau = 2k\pi \pm \Delta_\tau$ with $k \geq 0$, which define the length and direction of each bend is a bit more involved. Curves with central angles greater than 2π are not considered in this study, as they are more likely to describe the trajectory of a vehicle and not the geometry of a road segment. These angles can be calculated according to the following rule: if the curvature sign of a bend, as it was estimated from the previous step, matches that of the vertical component of the cross product between the previous and the next tangent vectors with respect to a right-handed coordinate system, then $\alpha_\tau = \Delta_\tau < \pi$.

Special consideration is required at bends of large radius, whose curvature sign may have been inaccurately estimated due to noise, for which the above assumption may not hold. The ambiguity is partly avoided by not allowing very small deflection angles in the first place so that the curvature sign can be reliably estimated using the tangents' direction vectors. In addition to this, the fact that this discrepancy occurs at arcs with a large radius, which in turn blend tangents with relatively small deflection angles, makes the estimation of the central angle value less ambiguous, as the total bend length estimated from the curvature plot can provide an accurate indication towards

the correct decision. In figure 8.9 for example, for a deflection angle as large as $\Delta_\tau = 30°$ and a radius as small as 300 m, the arc with a central angle $\alpha_\tau = \Delta_\tau$ is less than an order of magnitude shorter than that with a central angle $\alpha'_\tau = 360° - \Delta_\tau$. In conclusion, the central angle α_τ at each bend is equal to the deflection angle Δ_τ between the adjacent tangents, unless the curvature signs that is estimated from the curvature plot and the control polygon are opposite. In this case, the approximate bend length for the corresponding segment, estimated from the curvature plot, is compared to the arc lengths estimated using the radius from the least-squares fitting using $\alpha_\tau = \Delta_\tau < \pi$ and $\alpha'_\tau = 2\pi - \Delta_\tau > \pi$. The central angle that yields a total bend length closer to the approximate one is assigned to the corresponding control polygon vertex.

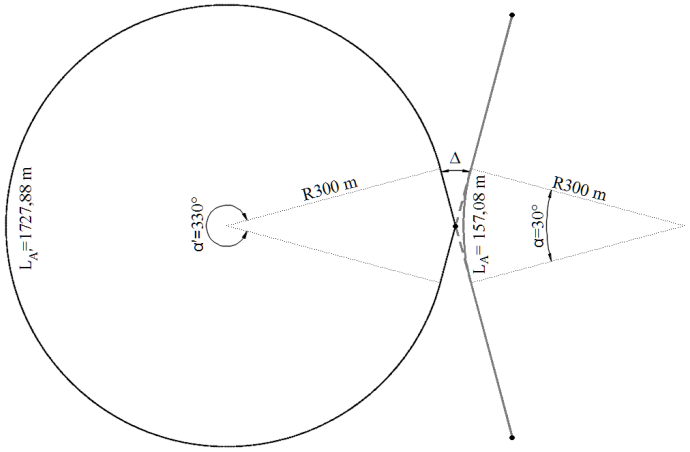


Figure 8.9: Estimating the central angle α at a control polygon vertex from the deflection angle Δ between the adjacent tangents.

8.2.3 Clothoid Spline Fitting

The parameter set for the construction of a clothoid spline comprises n circular arc segment radii R, n incurve and n outcurve clothoid segment lengths L^{in} and L^{out}, respectively, and $2(n + 2)$ coordinates $\boldsymbol{XY}^{cp}$ for the $n + 1$ control polygon vertices. The endpoints of each clothoid spline segment can be calculated from the central angles α_n for each bend and well known geometry formulae that can be found in a standard road design textbook (Kuhn, 2013). It is assumed that an asymmetric combined bend exists at each control polygon vertex, unless the radius is greater than a predefined threshold—2500 m in the examples presented herein—in which case a simple arc is used instead. As the endpoints of each segment and arc-length parameter for each input point are considered unknown, clothoid spline fitting is formulated as nonlinear optimisation problem (Piegl et al., 1997, p. 363).

As the parameter values were estimated in curvature space and independently refined in

object space from noisy data, the initial clothoid spline may not be close to the global minimum or even valid—meaning that the absence of overlapping segments is not guaranteed. In order to estimate a consistent parameter set, a clothoid-arc-clothoid sequence is independently fitted to each polygon vertex. In order to prevent overlaps, instead of the previous and next vertices of the control polygon, the midpoints of the adjacent tangents that were detected with RANSAC are used for each vertex, q.v. figure 8.7b. However, this may lead to a bend that does fit between the two tangent segments.

$$C_1^{cloth} = \sum_{q=1}^{m} \boldsymbol{d}_q^2 \qquad (8.13) \qquad C_2^{cloth} = \left(\frac{L_\tau^{in} + L_\tau^{out} - 2L_\tau^r}{L_\tau^{in} + L_\tau^{out}} \right)^2 \qquad (8.14)$$

such that

$$\frac{R_\tau^r}{3} \leq A_\tau^{in} = \sqrt{R_\tau^r L_\tau^{in}} \leq R_\tau^r \qquad (8.15) \qquad \frac{R_\tau^r}{3} \leq A_\tau^{out} = \sqrt{R_\tau^r L_\tau^{out}} \leq R_\tau^r \qquad (8.16)$$

where

C_1^{cloth}	=	proximity cost	C_2^{cloth}	=	bend segments' length cost
L_τ^{in}	=	in-curve clothoid segment length	L_τ^{out}	=	out-curve clothoid segment length
A_τ^{in}	=	in-curve clothoid parameter	A_τ^{out}	=	out-curve clothoid parameter
τ	=	bend index	n	=	total number of bends
q	=	sample point index	m	=	total number of sample points
R_τ^r	=	circular arc radius	L_τ^r	=	length of a circular arc segment
$\boldsymbol{d}_q$	=	distance from a sample point p to the candidate clothoid spline			

In order to override the problem of a potential invalid initialisation, a derivative-free multi-objective genetic algorithm (Deb, 2001) with two objective functions, q.v. equations 8.13–8.14, is used : $\underset{R^r, L^{in}, L^{out}, \boldsymbol{XY}^{cp}}{argmin} (C_1^{cloth}, C_2^{cloth})$. Equation 8.13 measures how well the clothoid spline fits the data; the fitting cost is defined as the sum of squares of the orthogonal distances $\boldsymbol{d}_q([X_q Y_q], clothoid(R^r, L^{in}, L^{out}, \boldsymbol{XY}^{cp}))$ between each point q and the candidate clothoid spline, thus overcoming the parameterisation problem (Pottmann et al., 2002). The second objective function, q.v. equation 8.14, aims to steer the solution towards one that combines the correct arc radius with clothoid lengths that are long and similar, as more than one radius-length pairs can fulfil the first objective. It is beyond the scope of this work to adhere to a specific set of road design guidelines, so only a basic set of rules are applied in the form of the constraints of equations 8.15

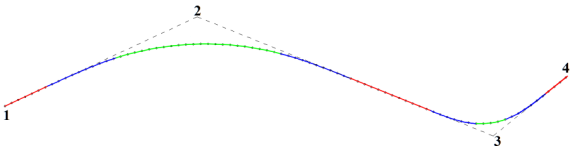
and 8.16 in order to guide the process to a reasonable outcome. From the set of Pareto-optimal solutions, the one with a radius value closest to the most probable radius in the set is selected.

Finally, the valid approximation of the full clothoid spline is used to initialise a gradient-based non-linear constrained optimisation scheme (Powell, 1978). The full parameter set is optimised using equation 8.13 as the objective function, with its derivatives approximated by central finite differences. The algorithm prototype was implemented using the Matlab (2017d) optimisation toolbox. The coordinates of each clothoid segment were evaluated using the implementation that was presented in Bertolazzi et al. (2010, 2014).

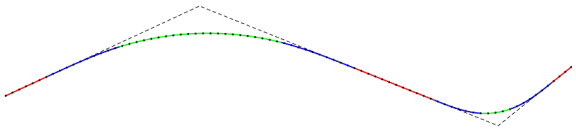
8.2.4 Experimental Results

The results of the proposed workflow on two datasets were compared to those acquired from the implementation of the method proposed in Baran et al. (2010), called Cornucopia (2010), and the "Best Fit Alignment" tool from the well-known road design software ©Autodesk AutoCAD Civil 3D (2018). Both datasets were created by sampling points along a clothoid spline and adding uniformly distributed noise along the normal direction. As Cornucopia (2010) only works with integer values for pixel positions, the coordinates of the sample points were rounded to 0.1 m and the experiments were conducted in decimetre units for consistency among all three methods. For the "Best Fit Alignment" tool, the default settings with respect to the desired and minimum circular arc radius to clothoid parameter ratio (R/A) were used—3 and 1, respectively—which might create an asymmetric clothoid as Cornucopia (2010) does not consider symmetric ones. No additional equality constraints on the in-curve and out-curve clothoid segment lengths were imposed for the proposed method, either, apart from the soft constraint of equation 8.14.

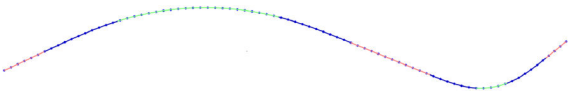
The first sample comprised equally spaced points at a 25 m interval along a 2 km long stretch with low curvature, q.v. figure 8.10a, and noise amplitude equal to 0.25 m. The input to Cornucopia (2010) was an ordered list of coordinates and an acceptable result was acquired using the default parameter set. The input to AutoCAD Civil 3D (2018) was a polyline—a piece-wise linear curve. For this example, the input curve was not resampled and the value for the maximum radius for the curve detection was set to R_{max} = 1500 m. As all three methods performed well with such datasets, q.v. figure 8.10, further investigations were conducted on a more challenging curve.



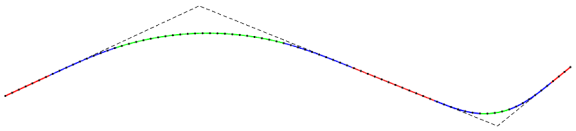
(a) Reference clothoid spline.



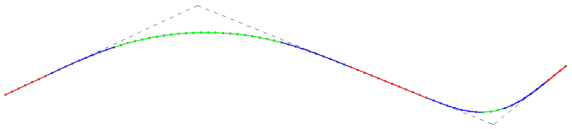
(b) The proposed method with D^{cloth} = 7.5 m and σ_n = 0.02 m



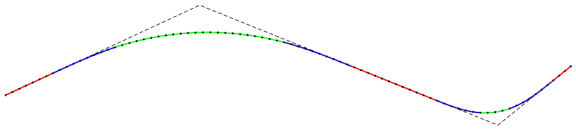
(c) Cornucopia (2010).



(d) The proposed method with D^{cloth} = 30 m and σ_n = 0.08 m.



(e) AutoCAD Civil 3D (2018) with R_{max} = 1500 m (without resampling).



(f) The proposed method with D^{cloth} = 60 m and σ_n = 0.16 m.

Figure 8.10: Clothoid spline fitting to a uniformly spaced sample (black dots) with 25 m sampling intervals and 0.25 m noise amplitude.

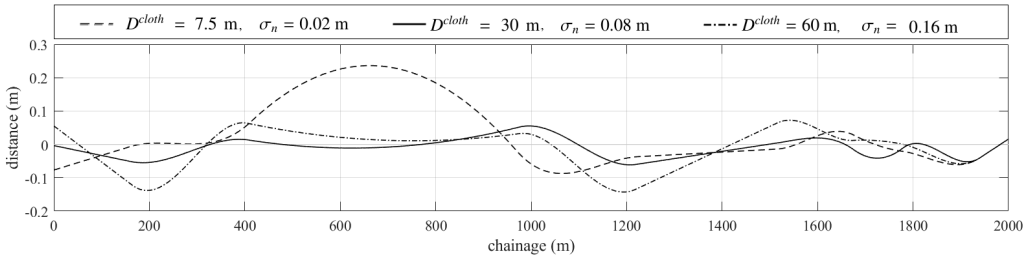


Figure 8.11: Horizontal alignment extraction - 1st Experiment: Signed orthogonal distances from the reference curve.

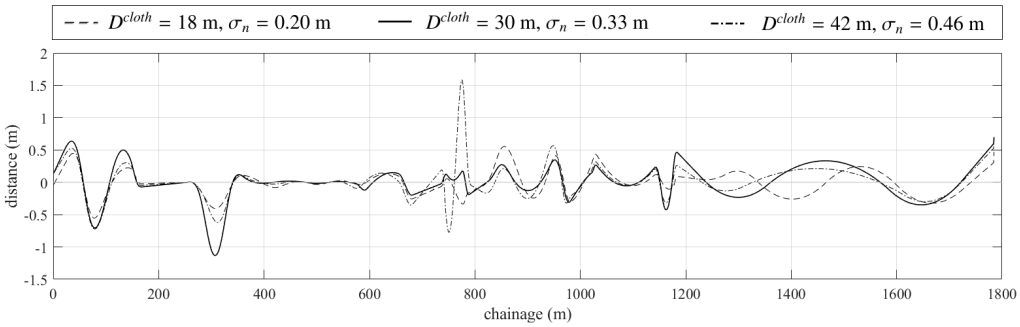
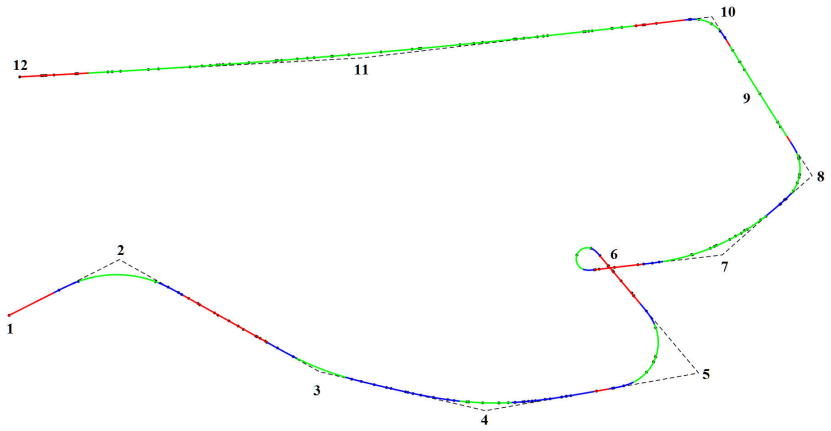


Figure 8.12: Horizontal alignment extraction - 2nd Experiment: Signed orthogonal distances from the reference curve.

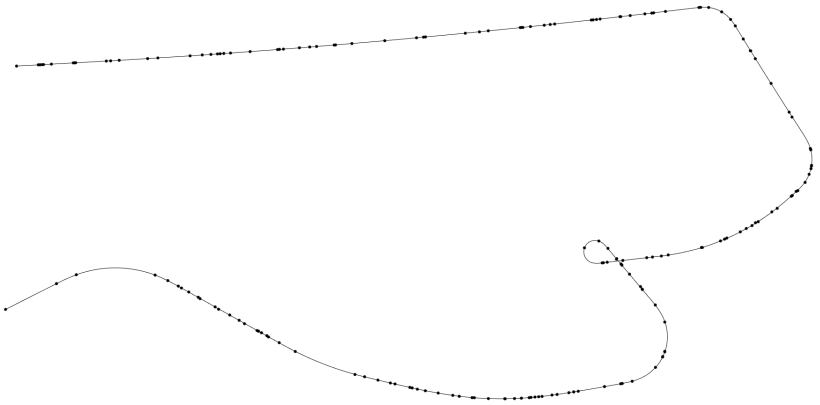
The second data sample was extracted at non-uniform intervals along an approximately 1.7 km long clothoid spline, q.v. figure 8.13a. The mean and standard deviation of the intervals along the reference clothoid spline were 10.03 m and 10.46 m, respectively; the noise amplitude was equal to 1 m. The arc radii at the bends vary from 10–8000 m and two tangents have zero length, posing a challenge to the curve segmentation stage.

Discovering a suitable set of parameter values in order for Cornucopia (2010) to return a reasonable result was not easy. Figure 8.14 shows one plausible solution which is a good fit but the primitive type sequence is not one that would be expected for a road or track centreline. Unfortunately, AutoCAD Civil 3D (2018) will not even process self-intersecting curves. When only the sample points alone were used, without explicitly providing the connecting edges, the intersection was simply ignored. In order to conduct the experiment regardless, the polyline was trimmed at vertex 6 and the bend was considered to have an apex angle lower than π. The tests were conducted using all combinations of the following values for the resampling rate $D_r \in \{1, 2, 5, 10, 15, 20, 25\}$ m and the maximum radius for curve detection $R_{max} \in \{25, 50, 200, 400, 600, 1000, 2500, 4000, 5000, 10000\}$ m. The best clothoid spline for the full length of the alignment was returned for $R_{max} = 1000$ m without resampling, see figure 8.16a. Due to the nature of the input, not every value pair returned a reasonable curve. The curve was then split in half; each segment was processed individually and returned results that were a better fit to the input curve. The results in figures 8.16b and 8.16c are indicative of the fact that R_{max} is used in a rather restrictive way.

The second point set was processed with the proposed algorithm using three different parameter pairs. Figure 8.5 shows the curvature-based segmentation result for $D^{cloth} = 30$ and $\sigma_n = 0.33$ m. No curvature zero-crossing was detected between the bends at vertices 3 and 4, which were split due to the peak detection step which, however, could not separate the bend at vertex 6 from the one at vertex 7; these were segmented by fitting the models in figure 8.7. For $D^{cloth} = 18$ m, $\sigma_n = 0.20$ the small neighbourhood size provided inadequate smoothing during the curvature estimation and the small noise standard deviation falsely justified an additional vertex where it did not exist. Bend 9 was ignored, as expected, because of the small deflection angle with respect to the noise level. The signed distances of each solution from the reference clothoid spline, q.v. figures 8.11 and 8.12, are within the envelope that is defined by the reference curve and the noise amplitude that was applied to the extracted point set, except along the sparsely sampled segments.



(a) Reference clothoid spline.



(b) The mean and standard deviation of the sampling intervals are 10.02 m and 10.45 m, respectively; the noise amplitude is equal to 1 m.

Figure 8.13: Horizontal alignment extraction - 2nd Experiment: Non-uniformly spaced sample.

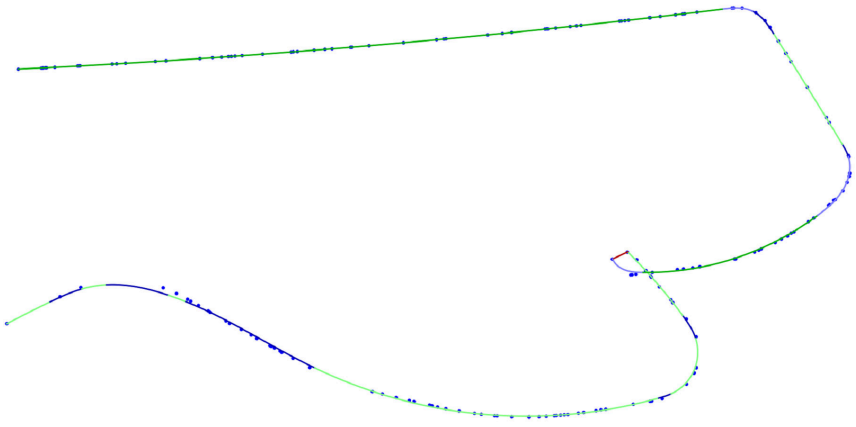
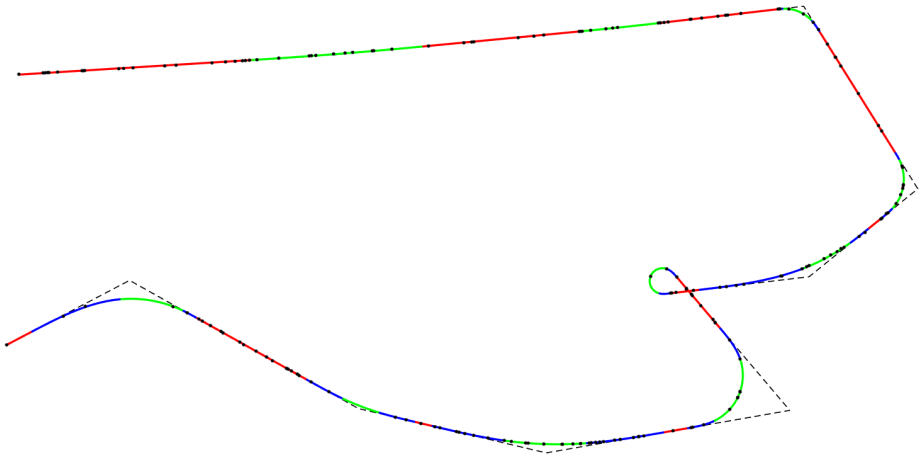
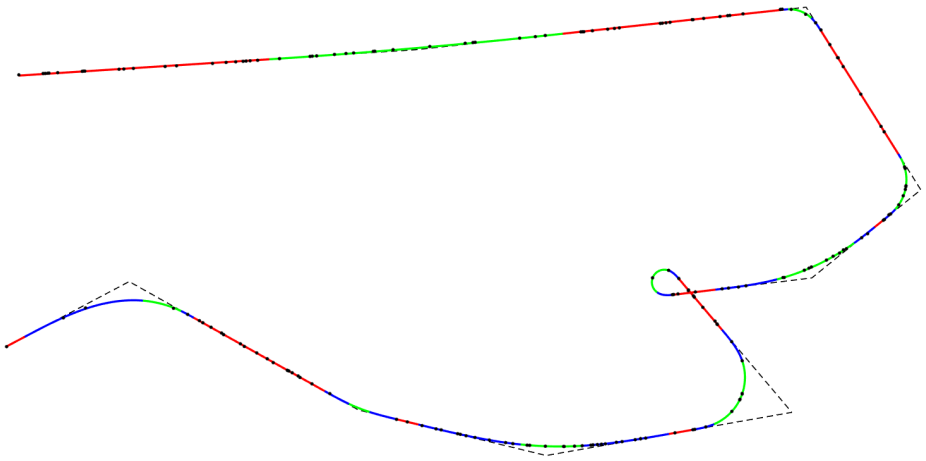


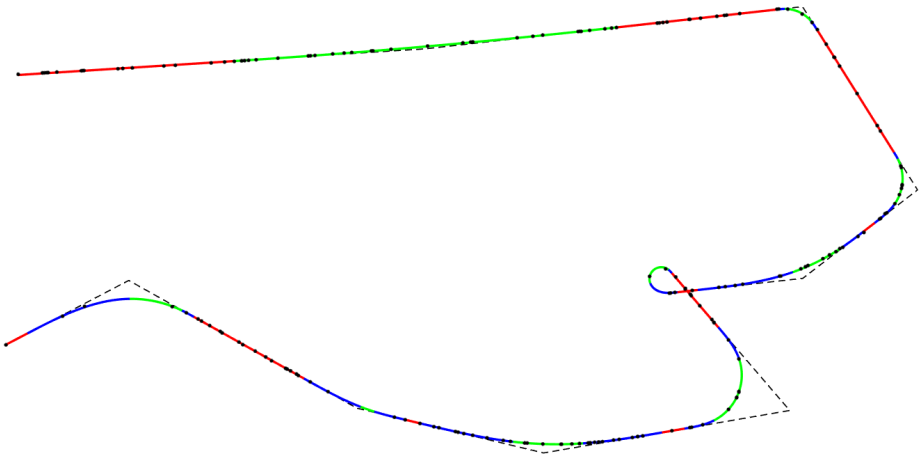
Figure 8.14: Horizontal alignment extraction - 2nd Experiment: Cornucopia (2010) on a non-uniformly spaced noisy sample, q.v. figure 8.13.



(a) D^{cloth} = 18 m, σ_n = 0.20 m.

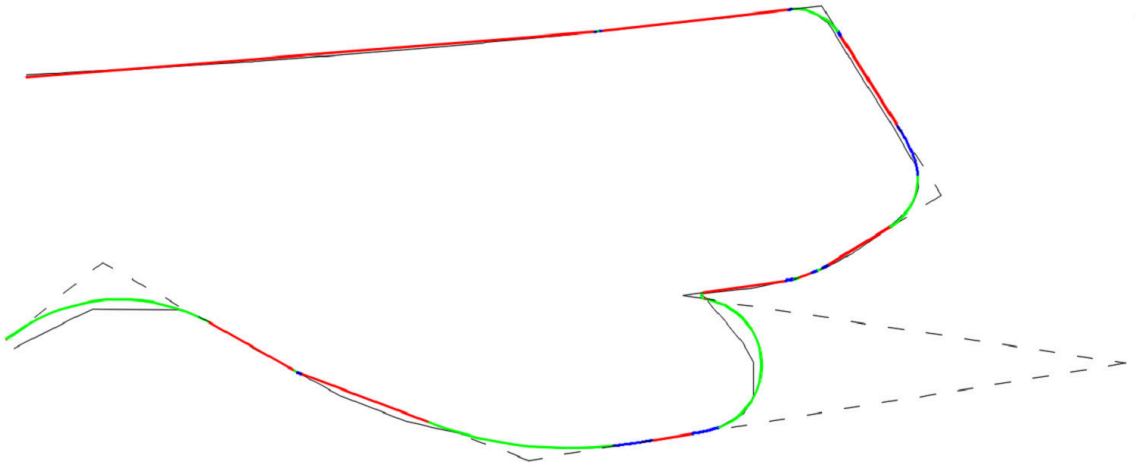


(b) D^{cloth} = 30 m, σ_n = 0.33 m.

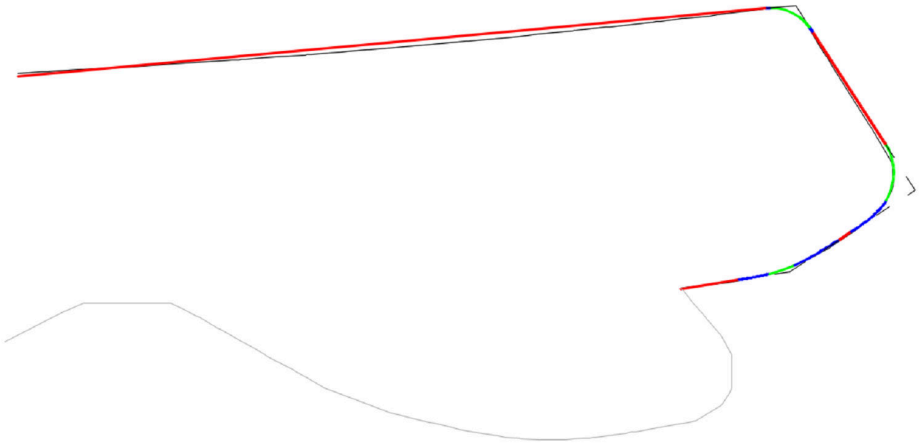


(c) D^{cloth} = 42 m, σ_n = 0.46 m

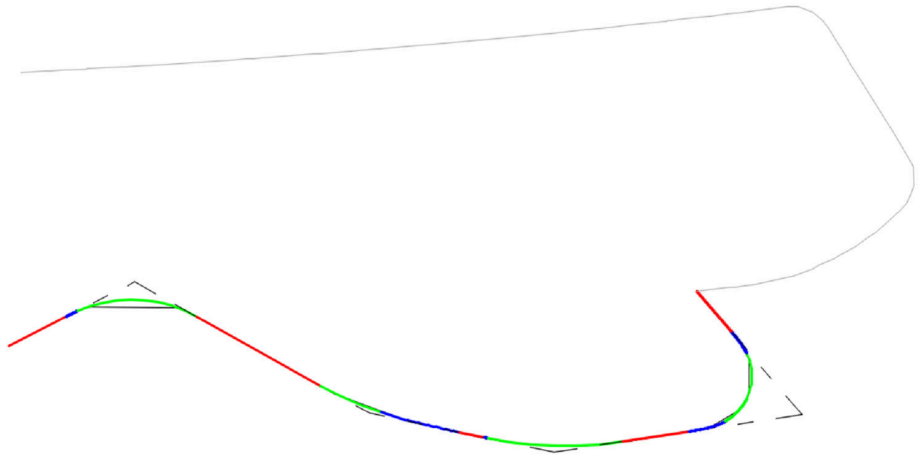
Figure 8.15: Horizontal alignment extraction - 2nd Experiment: The proposed method on a non-uniformly spaced noisy sample, q.v. figure 8.13.



(a) R_{max} = 1000 m without resampling.



(b) R_{max} = 200 m without resampling.



(c) R_{max} = 400 m without resampling.

Figure 8.16: Horizontal alignment extraction - 2nd Experiment: AutoCAD Civil 3D (2018) on a non-uniformly spaced noisy sample, q.v. figure 8.13.

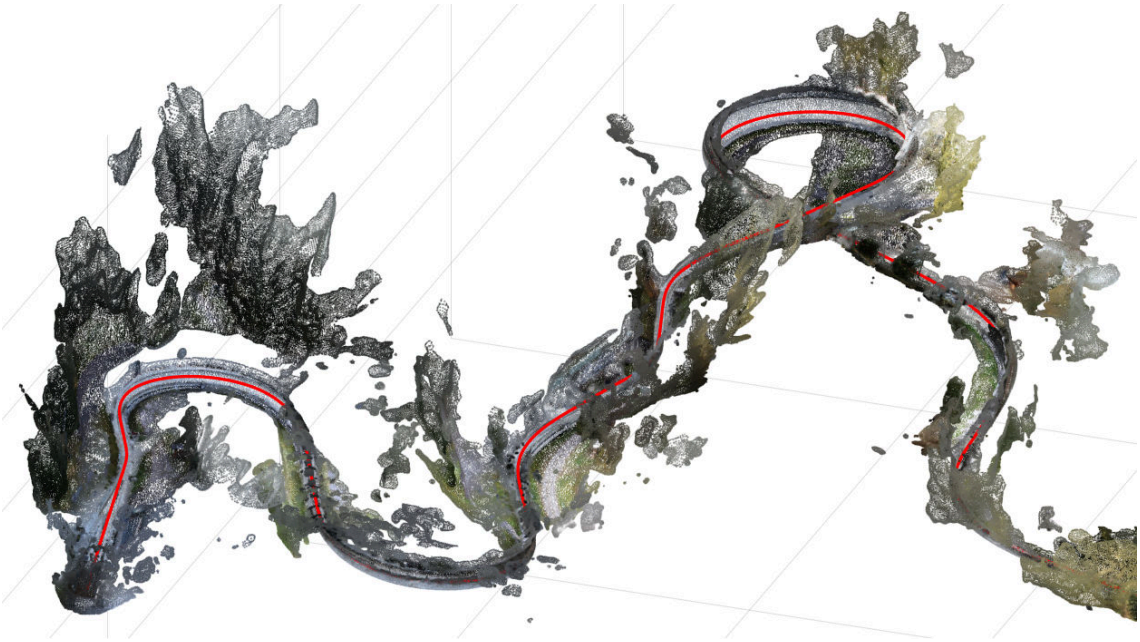


Figure 8.17: A 3D mesh which represents parts of the Igls track and its surrounding environment, as well as a space curve approximating its centreline alignment.

8.3 Track Surface Approximation

Given a 3D mesh that represents parts of the track and its surrounding environment, as well as a space curve that approximates its 3D centreline alignment, q.v. figure 8.17, a set of cross-sectional curves which can be used for the extraction of a parametric model of the complete extents of a track's surface is sought for. A sweeping-and-morphing approach based on a generalised cylinder representation, q.v. section 4.1, was deemed the most appropriate for track modelling for two reasons. Firstly, because the latter not only reflects the design intent of the infrastructure but it also correlates with the desired parameterisation of the track's surface using an along and across track parameter. Secondly, the fact that the track's surface geometry is decomposed to a space curve and cross-sectional plane curves, which in turn can be decomposed further to their constituent primitives, e.g. circular arcs or line segments, makes it easier to remove noise and clutter, fill in the missing information, enforce the required symmetry and regularity, and control the computational cost by selecting an appropriate along-track sampling interval.

Consider for example the straight segment of a track in figure 8.18, which can be represented in a simple way by a straight line parallel to the sidewalls and a single, almost rectangular, cross-sectional curve, q.v. figure 2.4d; sweeping the cross section along such a curve will yield the desired surface. However, a generic curve, e.g. a cubic smoothing spline, fitted to the sensor's

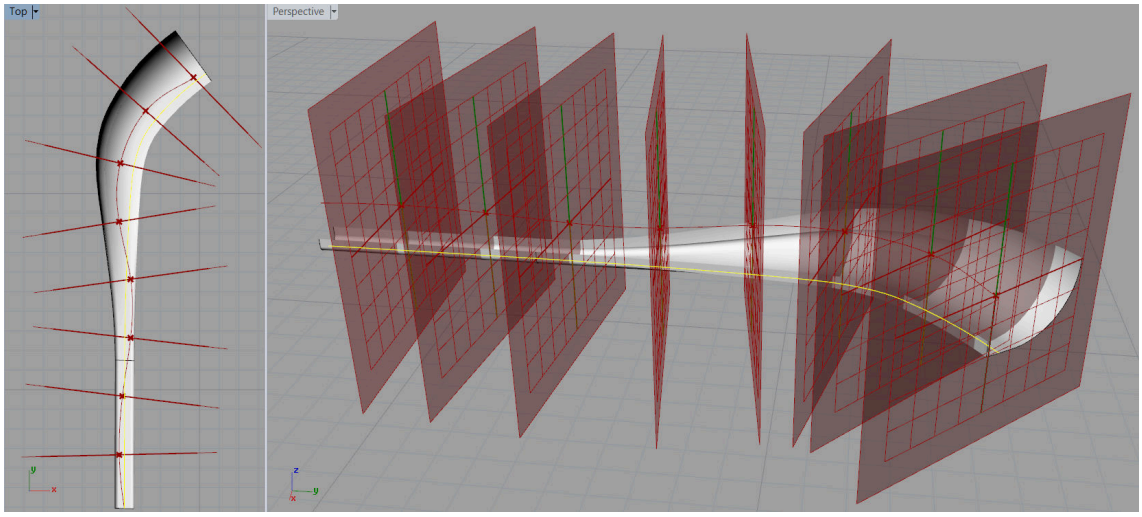


Figure 8.18: Extraction of cross-sectional curves along an approximate track alignment.

trajectory cannot be expected to be parallel to the sidewalls; optimising such a curve model by taking into account the 3D mesh of the track will be problematic if extraneous data, anomalies, or long along-track gaps exist, as its curvature cannot be easily controlled, q.v. section 4.2.

In addition, on a cross section that is not extracted perpendicular to a curve parallel to the sidewalls, the track floor cannot be expected to be horizontal—as it would also account for the along-track gradient—which is an attribute that can be used for inference and quality assessment purposes, q.v. section 8.1.2. Detecting the sidewalls' top edges and fitting a clothoid spline parallel to them, can provide a curve that closely follows the as-designed alignment, q.v. figure 2.4a, so that the extracted cross-sectional curves can be thought of as instances of the as-designed typical cross sections of the track, q.v. figure 2.4. However, the clothoid spline is a 2D structure; even though its geometry is flexible, if the input is referenced to a photogrammetric model with an arbitrary orientation with respect to a local horizontal plane, its rotation needs to be refined so that the projections of the locations that are extracted along the sidewalls correspond to that theoretical model. The problem becomes more challenging along the bends—compared to cylinder fitting—as the cross-sectional curves are not closed and do not have a fixed or standard geometry, q.v. figure 8.18.

The sweeping-and-morphing process begins with the extraction of cross sections from the mesh along an initial alignment, which is assumed to lie within the bounding volume of the track's surface. The track's sidewalls are detected on the cross-sectional curves by template matching, in order to create a best-fit clothoid spline at an approximately constant distance from them, close to the track's floor. Note that the objective is not to identify an optimal sliding trajectory down the track but rather use the track's geometric design intent, with respect to its curvature profile,

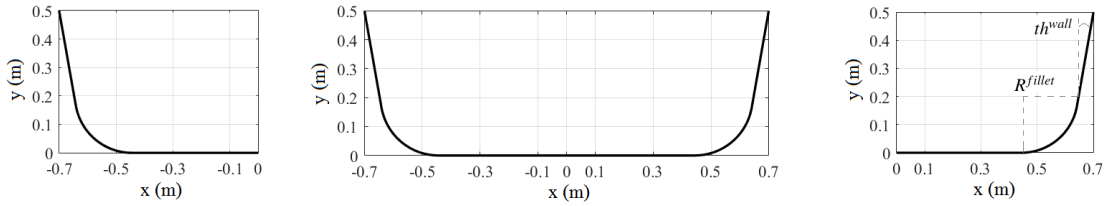
in order to infer its longitudinal shape. Finally, parts of the mesh where the track's surface is correctly reconstructed are detected, in order to extract a set of characteristic cross-sections and propagate them along the track's 3D alignment in order to fill in the missing parts.

8.3.1 Fitting the 3D Track Alignment

The initial track alignment is discretised by sampling m points at fixed length intervals—1 m intervals where used for the examples that are presented in this study. The coordinates at each station are denoted by $\boldsymbol{SC}_q^o = \boldsymbol{SC}^o(v_q) = [X_q^o, Y_q^o, Z_q^o]$ where v_q, with $q \in \{1, 2, \cdots, m\}$, is the 3D alignment's arc-length parameter; the arc-length parameter of its horizontal projection is denoted by t_q. At each station, a vertical cutting plane is used to extract cross-sectional curves from the mesh and define a left-handed 3D coordinate system centred at $\boldsymbol{SC}_q^o$ with the z axis parallel to the horizontal projection of the alignment's direction vector, the y axis vertical and the x axis horizontal. The 2D coordinates of each extracted section are denoted by $\boldsymbol{CS}_q^o = \boldsymbol{CS}^o(v_q) = [\boldsymbol{x}_q^o, \boldsymbol{y}_q^o] = [x_{q,i}^o, y_{q,i}^o]$, where $i \in \{1, 2, \cdots, a_q\}$ is the index of the a_q points that are sampled along each extracted cross-sectional curve for their discretisation and implicit representation—a sampling interval of 0.01 m was used for the examples which are presented herein with the points arranged in a counter-clockwise fashion. Cross-sectional curves that lie in their entirety above the sensor's path are ignored.

The sidewall detection along the straight and bend segments is based on typical full and half-section template curves, respectively, q.v. figure 8.19, which are generated based on user-defined attributes for the wall height h^{wall} and angle of inclination from the vertical th^{wall}, wall-floor fillet radius R^{fillet}, track width w^{track} at the straight segments, and the maximum width at the push-off stretch w^{push}; the values can be collected on site or default to 0.5 m, 5°, 0.2 m, 1.4 m (ice) or 1.5 m (concrete), and 3–4 m, respectively. The template coordinates are denoted by $\boldsymbol{CS}^T = [\boldsymbol{x}^T, \boldsymbol{y}^T]$ with $\boldsymbol{x}^T = x_j^T$ and $\boldsymbol{y}^T = y_j^T$ where $j \in \{1, 2, \cdots, b^T\}$ is the index of the b^T points that are sampled along each template; the superscript $T \in \{left(TL), full(TF), right(TR)\}$ represents all three template types. The origin of each template's 2D coordinate system is assumed to coincide with the track's actual alignment and the highest points with the sidewall edges.

The translation $\boldsymbol{T}_{xy}^{T_0}$ between all the candidate template curves $\boldsymbol{CS}^T$ and each one of the continuous cross-sectional plane curves $\boldsymbol{CS}_q^o$ is initialised by a template matching process after converting them to raster format using a $sr \times sr$ grid with a spatial resolution of $sr = 7.5cm$, which is essentially an estimation of the noise amplitude. The values for sr and th^{wall} are selected so that the process is robust to any discrepancies in the track width or inaccurate datum definition. Template



(a) Left half-section. **(b)** Straight segment full section template. **(c)** Right half-section.

Figure 8.19: Examples of typical track cross-sectional curves.

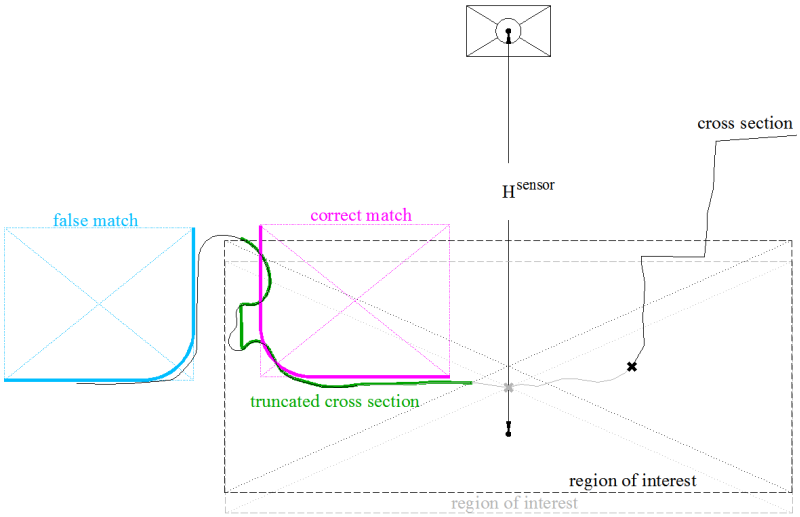


Figure 8.20: Defining the floor level on a cross-sectional curve and the region of interest for template matching.

matching is based on the sum of absolute differences and restricted to a $1.5w^{track} \times 2h^{wall}$ region of interest, in order to avoid false matches, q.v. figure 8.20. For the first 50 m of the track, along the starting straight, the region's width is set to $1.5w^{push}$ and the width of the full template is adjusted according to the wall-to-wall distance of the already translated half-section templates. Cross sections whose rasterised bounding box dimensions are smaller than the template's, are ignored. In the general case, the region of interest is centred at the origin of the cutting plane's coordinate system, however, if the sections are extracted along the sensor's path—which is assumed to be at a constant height H^{sensor} above the track floor—the region of interest is centred at the vertical projection of the origin on the cross-sectional curve. If a station lies above a mesh gap, the vertical difference between the cross section vertex closest to the point that lies H^{sensor} m below the origin is used to define the region's centre, q.v. figure 8.20.

Each cross-sectional template curve—in vector format—is translated according to the location of the matched raster template's centre. As this initial translation $\boldsymbol{T}_{xy}^{T_0}$ is too coarse, the transformation of the template is refined in a least-squares fashion by taking into account the segment of the

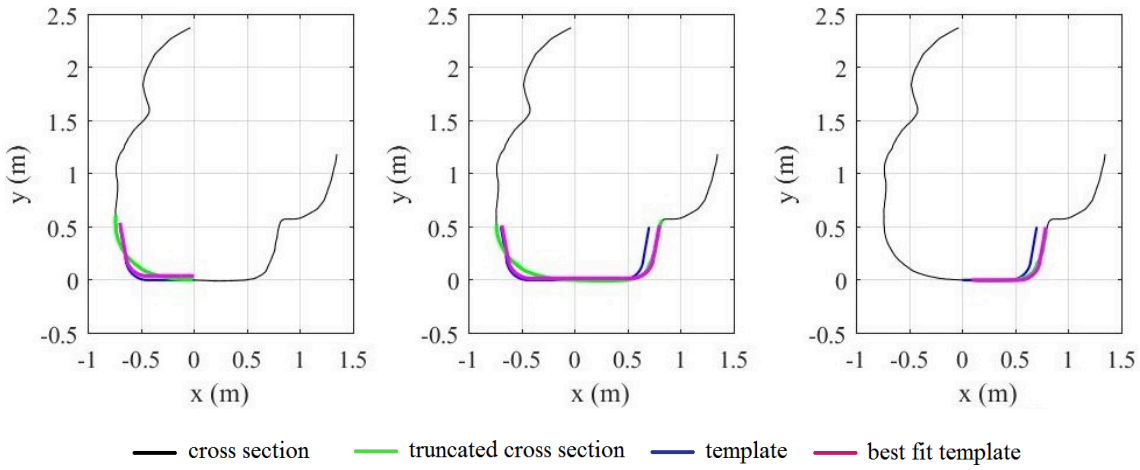


Figure 8.21: Template matching on a cross section that was extracted along the entrance of a right bend. The best match is shown in figure 8.23c.

cross-sectional curve close to the template. The extracted cross sections are truncated by selecting the points within $3sr$ from the translated template curve, q.v. figure 8.20. If the length of the truncated cross section $\boldsymbol{CS}^{o'}$ is less than half the template length or gaps greater than $3sr$ exist, the template is considered to be an invalid match. The translation $\boldsymbol{T}^{T}_{xy}$ of each valid candidate template match $\boldsymbol{CS}^{T'}$—and for the full cross sections also the scale S_x along the X-axis, i.e. the width—is optimised by minimising the sum of squares of the orthogonal distances $\boldsymbol{d}(\boldsymbol{CS}^{o'}, \boldsymbol{CS}^{T'})$ from each truncated cross-sectional curve point to the template; the distances for points closer than sr from the lowest part of the translated template, i.e. the floor, are weighted by a factor of $\mu = 2$:

$$\underset{\boldsymbol{T}_{xy}, S_x}{argmin} \sum_{ii=1} \mu_i \boldsymbol{d}^2_{ii}\Big(\boldsymbol{CS}^{o'}_{q}, S_x(\boldsymbol{CS}^{T'} + \boldsymbol{T}^{T_0}_{xy}) + \boldsymbol{T}_{xy}\Big) \tag{8.17}$$

where $\boldsymbol{CS}^{o'}_{q} = \boldsymbol{CS}^{o}_{q,ii}$ with $\boldsymbol{d}(\boldsymbol{CS}^{o}_{q,ii}, \boldsymbol{CS}^{T} + \boldsymbol{T}^{T_0}_{xy}) < 3sr$, $ii \subseteq \{1, 2, \cdots, a_q\}$, and $\mu_{ii} = 2$ if $\boldsymbol{d}(\boldsymbol{CS}^{o}_{q,ii}, \boldsymbol{CS}^{T}_{jj} + \boldsymbol{T}^{T_0}_{xy}) < sr$ with $y^{T}_{jj} = 0$ and $jj \subset j$, or $\mu_i = 1$ otherwise.

The selection of the best match template $\boldsymbol{CS}^{T}_{q}$ between the valid full, right, or left candidates $\boldsymbol{CS}^{T'}$ along the track, beyond the starting straight, is performed as follows: the full template is selected if all three templates are a valid match and the RMS for the left and right half-section templates is less than $0.3sr$ or the ratio of the smallest to the greatest between them is below 0.8. If the full template is rejected, the left or right half-section template with the smallest $RMS(\boldsymbol{d})$ is selected as the best match.

In order to define the track's alignment with respect to a physical reference at a fixed offset from the sidewalls, the template points that correspond to the track's centreline are transformed back to 3D space and their projections $[X^T(t_q), Y^T(t_q)]$ on the horizontal plane are used to fit an approximating clothoid spline according to the automatic process that was described in q.v. section

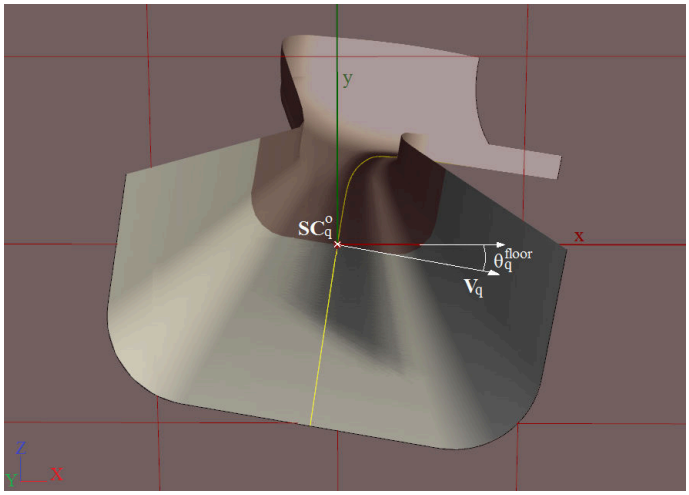


Figure 8.22: A vertical cross-sectional cutting plane (highlighted in red) along the alignment. SC_q^o is the point along the alignment that defines the origin of the cross section's coordinate system. The rotation θ_q^{floor} of the floor's direction vector V_q with respect to the x-axis should be equal to zero.

8.2. For the vertical profile of the alignment, the elevation of each template point with respect to its position along the clothoid spline $Z^T(t_q)$ is smoothed by a robust local weighted regression scheme (LOESS) with a 2nd degree polynomial (Cleveland et al., 1988; Matlab, 2017g) and a span equal to $0.5D^{cloth}$. An approximating cubic spline (Matlab, 2017f) with 1 polynomial segment for every 10 m of the clothoid length is fitted to the function $g\colon t_{qq} \mapsto Z_{qq}$, with $qq \subseteq \{1, 2, \cdots, m\}$ and $rsd^{profile}(t_{qq}) = |Z^{smooth}(t_{qq}) - Z(t_{qq})| < 10median(rsd^{profile})$ where $rsd^{profile}(t_{qq})$ are the residuals between the input $Z(t_{qq})$ and smoothed $Z^{smooth}(t_{qq})$ coordinates. The detection process is performed again on the cross sections that are extracted along the fitted 3D alignment of the track.

If the datum of the input was only approximately known—e.g. the mesh and 3D alignment were referenced with respect to the coordinate system of the photogrammetric model—the widths and floor gradients, q.v. figure 8.22, of the fitted templates are used within the optimisation scheme that was described in section 8.1.2, q.v. equations 8.6 and 8.7, and the process is repeated. In this case equation 8.17 becomes

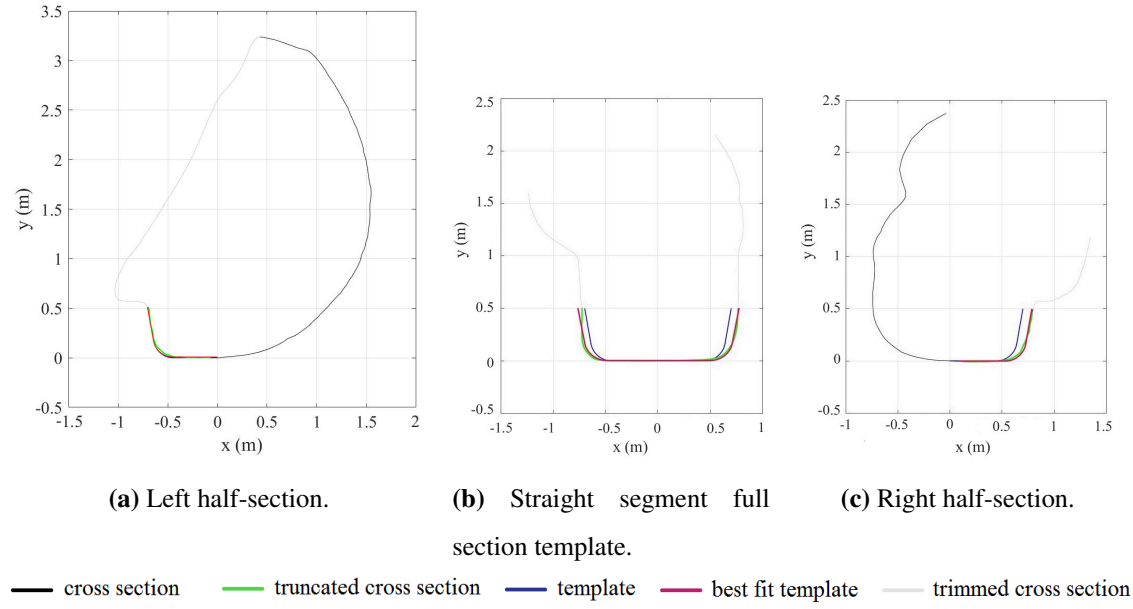
$$\underset{\boldsymbol{\theta}^{floor},\boldsymbol{T}_{xy},S_x}{argmin} \sum_{ii=1} \mu_j \boldsymbol{d}^2\Big(\boldsymbol{CS}^{o'}q, \boldsymbol{R}_{xy}S_x(\boldsymbol{CS}_j^{T'} + \boldsymbol{T}_{xy}^0) + \boldsymbol{T}_{xy}\Big) \tag{8.18}$$

where $\boldsymbol{R}_{xy} = \begin{bmatrix} cos(\theta^{floor}) & -sin(\theta^{floor}) \\ sin(\theta^{floor}) & cos(\theta^{floor}) \end{bmatrix}$ and the best match selection process is adjusted so that the full template is rejected if the absolute difference for the left and right half-section template θ^{floor} is greater than 5°.

8.3.2 Cross-Sectional Curve Segmentation

At this point, a set of cross-sectional curves $\boldsymbol{CS}_q^o$, which were extracted along a 3D alignment approximately parallel to the track's floor and sidewalls, are available. In addition, a template curve $\boldsymbol{CS}_q^T$ is fitted to a subset of the extracted cross sections. The shape of the final set of cross sections $\boldsymbol{CS}_q$ is acquired by removing (trimming) the parts of the input cross sections that do not correspond to the track's surface. Firstly, those cross sections on which a template was detected are trimmed at the points closest to the ones that represent the top of the sidewall on each template, q.v. figure 8.23. Any part that lies farther away than 2 m or 5 m—the former is the nominal track width plus the wall height and the latter is an estimation of the maximum height of the track's wall along a bend—depending on whether a template was detected or not, along the curve from the origin on either side is also removed, in addition to any bounding box restrictions during the mesh reconstruction. If the cross section is a closed curve, the point with the highest y value is also used as a cutting point, q.v. figure 8.23a. Then, each cross-sectional curve is trimmed as follows, q.v. figure 8.24:

- the cross-sectional curve is simplified using the Douglas et al. (1973) algorithm, with $tol_{DP} = sr$
- the counterclockwise angle θ^n between the positive x-axis and the normal direction of each segment is estimated
- The arc length parameter s_{p^N}—along the input curve—of the endpoints of each segment with $-120° < \theta^n_{p^N} < -60°$ are selected as cutting point candidates
- the discrete curvature κ is estimated at each point based on the osculating circle, using the two neighbouring points—one on either side—q.v. section 8.2.1
- The arc length parameters of local extrema s_{p^E} of κ are added to the candidate list
  - candidates with $\kappa_{p^E} < 1\ m^{-1}$ are removed from the list
- candidates with y coordinate less than twice the wall height $y_{p^E} < 2 * h^{wall}$, are removed from the list
- The two extrema closest along the curve to the vertical projection of the origin on the cross section, one on either side, are selected as the cutting points $min(|s_p - s_o|)$



(a) Left half-section. **(b)** Straight segment full section template. **(c)** Right half-section.

Figure 8.23: Trimming the cross-sectional curves at the points which are closest to the best-match template endpoints that correspond to the sidewall edges.

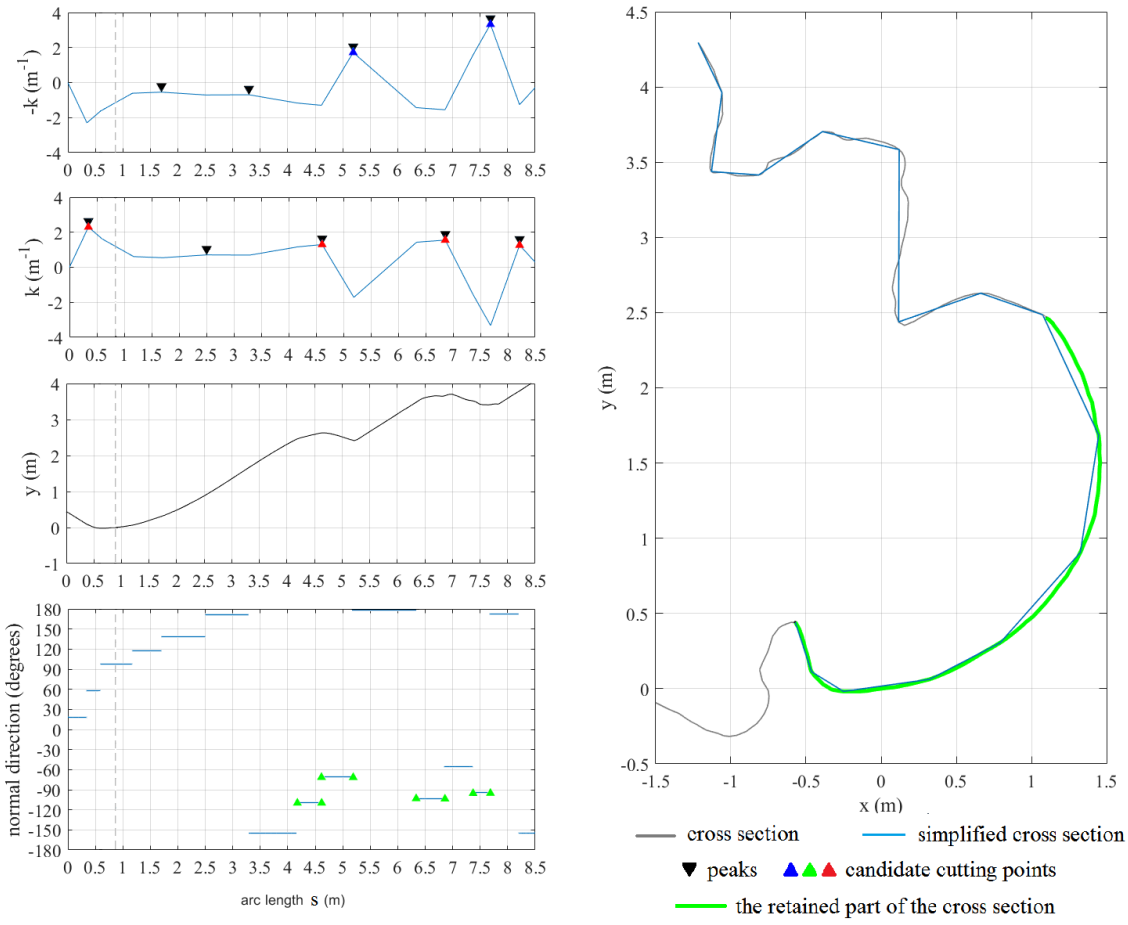


Figure 8.24: Shape-based cross-sectional curve segmentation.

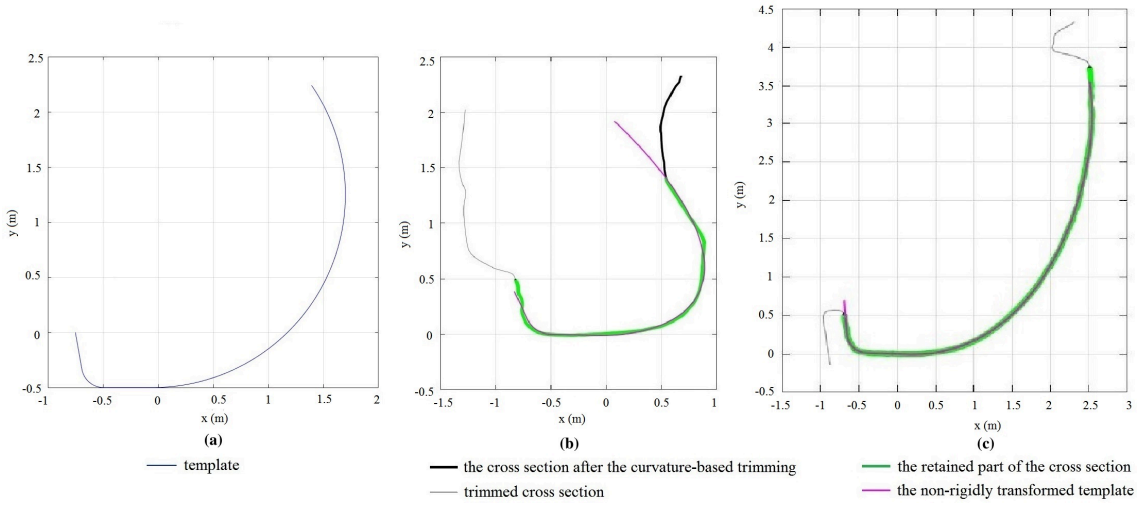


Figure 8.25: Two examples of CPD-based curve segmentation. The same generic template curve is used for all sections along left bends.

Table 8.1: Coherent point drift algorithm parameter settings for non-rigid registration of typical cross-sectional curves to those that were extracted along a track.

•cross section segmentation				
•characteristic straight segment cross-sectional curve extraction				
iterations	normalise	outliers	β	λ
100	no	95%	10	2
•characteristic bend cross-sectional curve extraction				
iterations	normalise	outliers	β	λ
100	no	1%	5	2

The upper edge of the track's surface may not appear as a sharp corner either because it was not constructed as such—e.g. the wooden guard acting as a sidewall for the sled is far from being perpendicular to the track's surface—or because it was smoothed during the mesh generation. The CPD algorithm, q.v. section 4.1, is used with each cross section where a half-section was detected, in order to non-rigidly transform a generic bend cross section template to the cross section, q.v. figure 8.25a; the algorithm's parameter settings are given in table 8.1. The cross section is trimmed at the highest point where the two curves begin to diverge, q.v. figure 8.25b—which is estimated as the local maximum of the curve-curve distances—unless the maximum distance or the distance between the upper endpoints is less than sr. As most parts of each cross section that do not correspond to the track's surface have been removed, this approach is expected to remove the remaining parts without oversegmenting the curve, q.v. figure 8.25c.

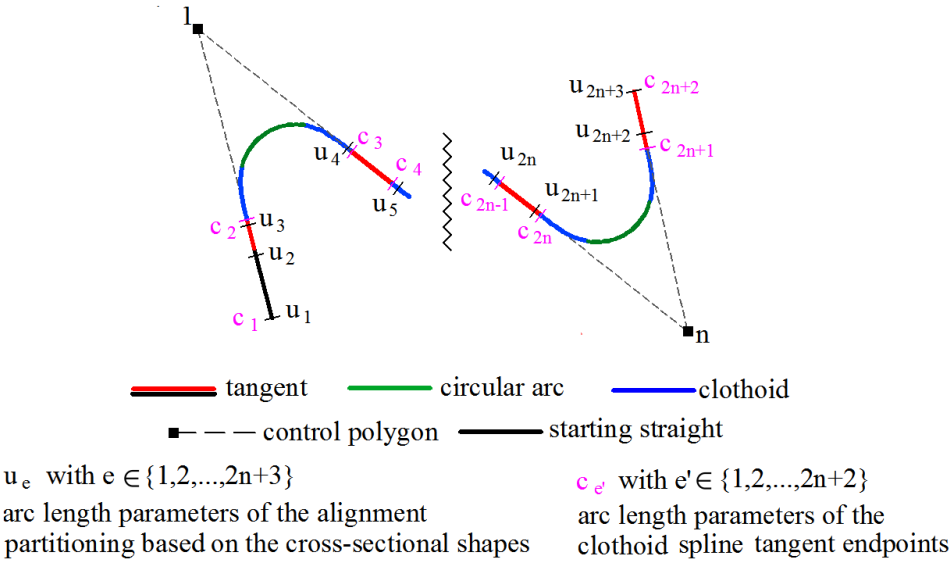


Figure 8.26: Track partitioning for the skinning process.

8.3.3 Sweeping and Morphing

The sweeping-and-morphing algorithm is a shape detection and optimisation process that is applied to each track bend and straight segment independently and aims to identify the boundaries and generate an initial approximation of the full extents of the track's surface. This is accomplished by interpolating between characteristic cross-sectional curves that have been reconstructed successfully, in order to fill gaps along and across the track's surface.

An alignment with n bends—and the set of m cross sections CS_q which were extracted along its length and subsequently segmented, q.v. section 8.3.2—is partitioned along the clothoid spline's straight segments into $2n+2$ parts. The arc length parameters u_e, with $e \in \{1, 2, \cdots, 2n+3\}$ and $u \in t$, of the endpoints of each part, q.v. figure 8.26, are defined by the first and last cross sections along each clothoid spline tangent where a full template was detected, q.v. figures 8.19b and 8.27. Each part is modelled by the corresponding 3D alignment segment, the set of trimmed cross sections acting as the observations of the target geometry, a set of characteristic cross sections, and a morphing function $f^m\colon [0,1] \mapsto [0,1]$ which is used in order to interpolate between the shapes of the latter along the path. The arc length parameters v_q of the stations along each path segment with $u_e \leq t_q \leq u_{e+1}$ are normalised according to $\bar{v} = \dfrac{v_q - v_{q0}}{v_{q1} - v_{q0}} \in [0,1]$, where v_{q0}, v_{q1} are the arc length parameters of the segment's endpoints, $t_{q0} = u_e$ and $t_{q1} = u_{e+1}$.

The generic cross-sectional template curves were useful for detection and segmentation, however, a data-driven approach based on the extracted cross sections is required in order to generate a characteristic cross section for the straights and for each bend and use it in the fitting process.

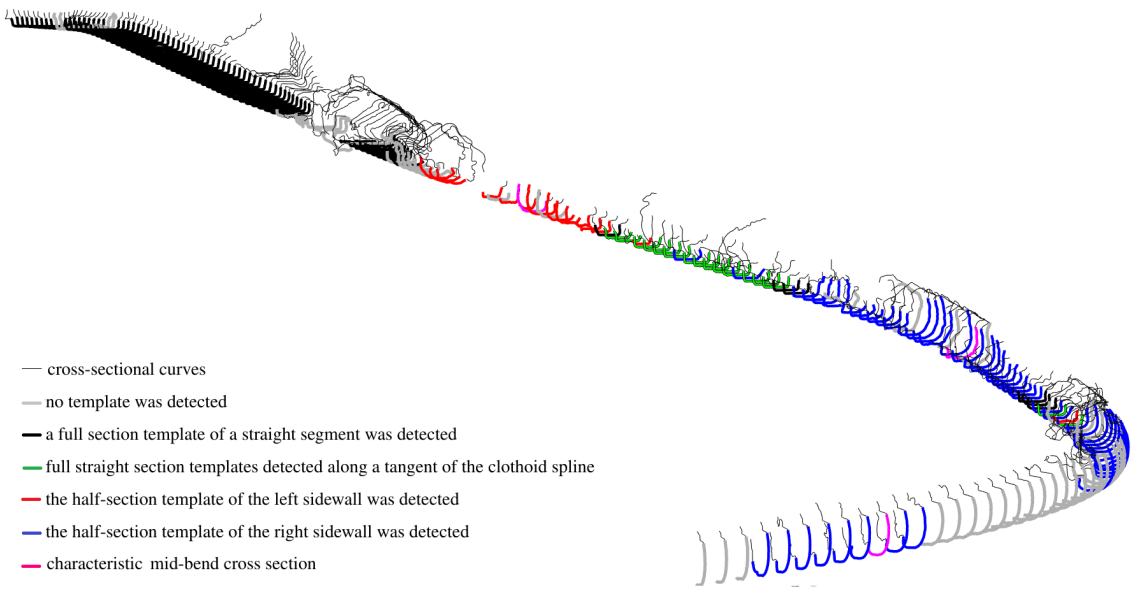


Figure 8.27: Cross-sectional curve detection, segmentation, and selection along the Lillehammer track.

Figure 8.29 shows the geometric features of each cross-sectional curve, for a set of trimmed cross-sectional curves, with respect to their position along the clothoid spline and its curvature, which are used in order to generate the characteristic sections, where:

- length $L_q^{CS} = \sum_{i=1}^{a_q-1} \sqrt{(x_{q,i+1} - x_{q,i})^2 + (y_{q,i+1} - y_{q,i})^2)}$
- bounding box dimensions, i.e. width w^{CS} and height h^{CS}
  - $w_q^{CS} = max(\boldsymbol{x}_q) - min(\boldsymbol{x}_q)$
  - $h_q^{CS} = max(\boldsymbol{y}_q) - min(\boldsymbol{y}_q)$
- endpoint height difference $dy_q^{CS} = y_{q,a_q} - y_{q,1}$
- the median $medx_q$ of the x components of the normal direction vector at each point along the cross section
- normalised along-track distance from the bends' midpoints $d\bar{t}m_q$

In order to generate the characteristic cross section for the straight segments of the track, all the cross sections with:

$$0.9w^{track} \leq w_q^{CS} \leq 1.1w^{track}, \tag{8.19}$$

$$0.9h^{wall} \leq h_q^{CS} \leq 1.1h^{wall}, \text{and} \tag{8.20}$$

$$0.9L^{fs} \leq L_Q^{CS} \leq 1.1L^{fs}, \text{where } L^{fs} = w^{track} + 2h^{wall} + (\pi - 4)R^{fillet} \tag{8.21}$$

are stacked together and used as the target geometry in order to non-rigidly transform the full section template, q.v. figure 8.19b, using the CPD algorithm, q.v. figure 8.25. As even the slightest across track slope at the starting area, where the sled's speed is low, may affect its trajectory, a template with the nominal push-off line width is used as the typical cross section at the start of the alignment. In reality, this is prevented by two grooves which are created on the ice along the push-off straight, figure 7.22.

In order to select a characteristic section for each bend, the function $g\colon t_q \mapsto {L_q^{CS}}^2 + {w_q^{CS}}^2 + {h_q^{CS}}^2 + {dy_q^{CS}}^2 + {medx_q}^2 + \bar{dtm}_q^2$ is smoothed by a robust local weighted regression scheme (LOESS) with a 2nd degree polynomial and a span equal to $0.5D^{cloth}$ (Cleveland et al., 1988; Matlab, 2017g), and its local maxima $[t_q, p_q^g]$ are detected. The cross section $\boldsymbol{CS_\tau} = \boldsymbol{CS}_q$ with $|rsd_q^{CS}| < 3median(|rsd^{CS}|)$—where $rsd^{CS} = smooth(g) - g$ are the residuals of the smoothing process and $\tau \in \{1, 2, \cdots, n\}$ is the index of the bend—which is closest along the track to the highest peak $max(p^g)$ within each interval $(u_{2\tau+1}, u_{2\tau+2})$, is used as the target geometry in order to non-rigidly transform the corresponding template—q.v. figure 8.25a—using the CPD algorithm and generate the typical cross section for the corresponding bend. The parameter settings for CPD are given in table 8.1; it is assumed that most outliers have been removed so the flexibility of the input curve is increased by decreasing the value for β from 10 to 5. If a left or right half-section was not detected on the selected section, the inner half-section is replaced by a left or right half-section template, respectively, q.v. figure 8.28 (upper left). The cross sectional curves which where not considered due to their inconsistent shape are assigned a weight factor η_q in order to decrease their influence during the optimisation process according to:

$$\eta_q = \begin{cases} 0.5, \text{ if } |rsd_q^{CS}| > 3median(|rsd^{CS}|) \\ 1, \text{otherwise} \end{cases} .$$

The push-off line, straight, and mid-bend typical cross sections are represented by $\boldsymbol{CS}^P = [\boldsymbol{x}^P, \boldsymbol{y}^P] = [x_j^P, y_j^P]$, $\boldsymbol{CS}^S = [\boldsymbol{x}^S, \boldsymbol{y}^S] = [x_j^S, y_j^S]$, and $\boldsymbol{CS}_\tau^B = [\boldsymbol{x}_\tau^B, \boldsymbol{y}_\tau^B] = [x_{\tau,j}^B, y_{\tau,j}^B]$, respectively, with $j \in \{1, 2, \cdots, b^m\}$ which denote the equal number $b^m = 30$ of control points of an approximating quadratic spline (Matlab, 2017f) fitted to the characteristic cross sectional curves, q.v. figure 8.30c. The cross sections along the entire length of each track segment, $\boldsymbol{CS}^m(\bar{v}) = [\boldsymbol{x}^m(\bar{v}), \boldsymbol{y}^m(\bar{v})] = [x_j^m(\bar{v}), y_j^m(\bar{v})]$, are obtained by interpolating between these control points and using the interpolated control points to construct a quadratic spline for each one. This operation is denoted by $spl(\boldsymbol{CS}_{P,S,B}^m)$; for the discretisation of each spline, in order to estimate the curve-to-curve distances as a metric for their similarity in equation 8.25, it is assumed that the number of points is equal to those of corresponding cross-sectional curve $\boldsymbol{CS}$.

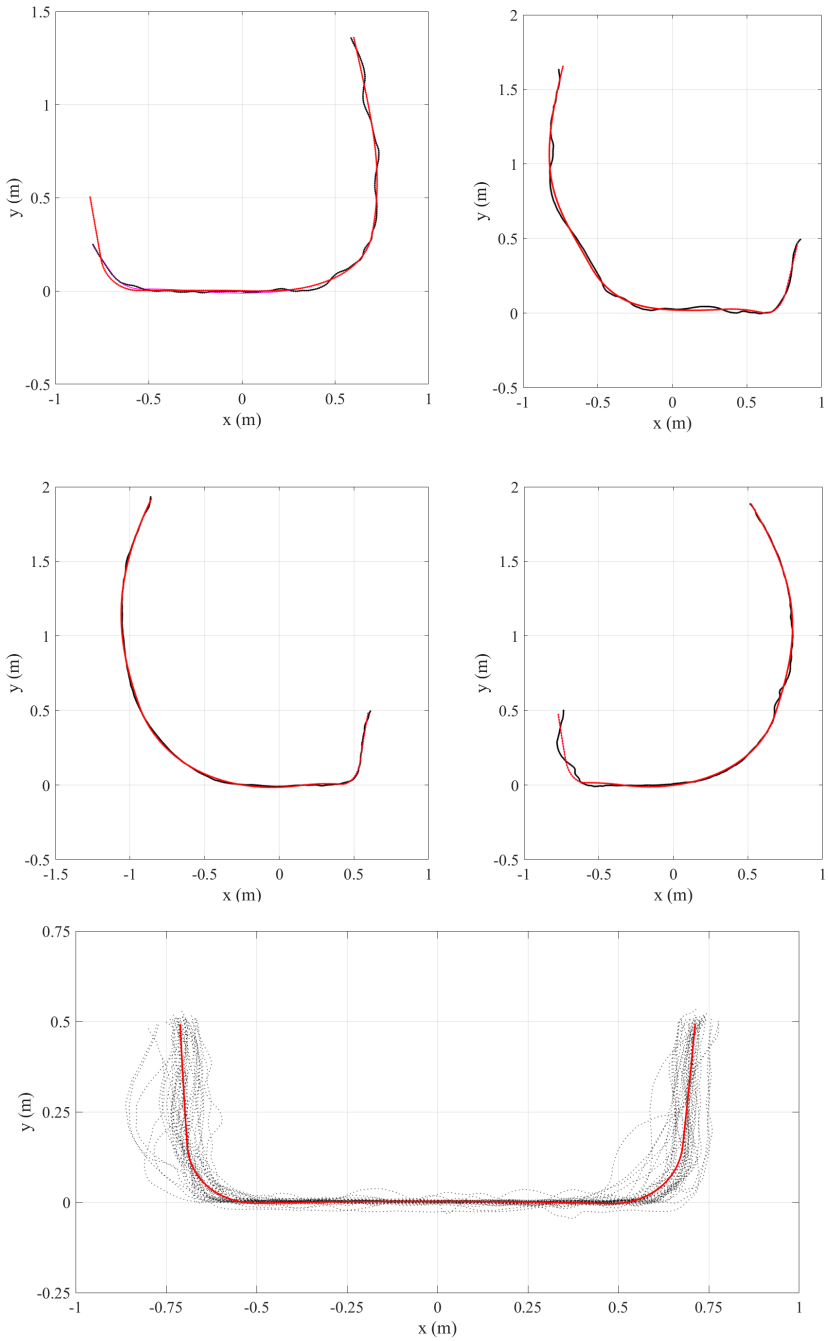


Figure 8.28: Data-driven generation of the characteristic cross-sectional curves for the first four bends and the straights along the Lillehammer track, extracted from a photogrammetric point cloud, q.v. section 10.4.

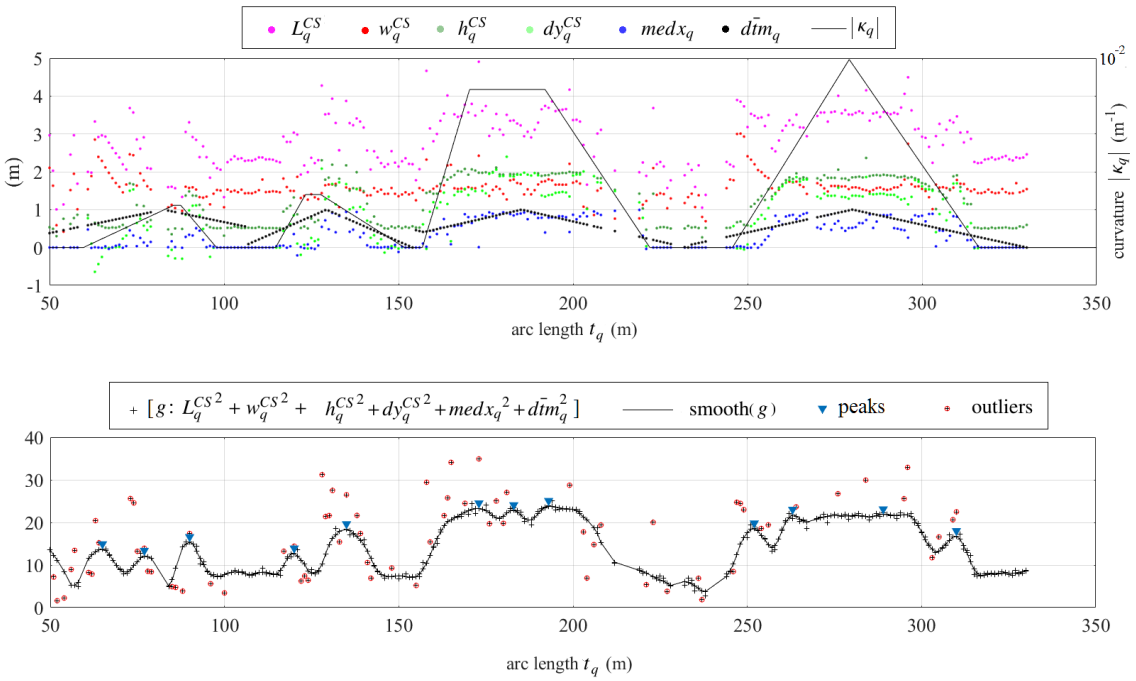


Figure 8.29: The geometric properties of the trimmed cross-sectional curves' shape that are used for the characteristic section selection at each bend.

The interpolation is calculated according to:

- along the straight segments

$$\boldsymbol{CS}_S^m(\bar{v}) = (\boldsymbol{CS}^S + \boldsymbol{D}^{xy}(v_{q0}))(1-\bar{v}) + (\boldsymbol{CS}^S + \boldsymbol{D}^{xy}(v_{q1}))\bar{v} \tag{8.22}$$

- along the starting straight

$$\boldsymbol{CS}_P^m(\bar{v}) = \boldsymbol{CS}^P(1 - f_P^m(\bar{v})) + (\boldsymbol{CS}^S + \boldsymbol{D}^{xy}(v_{q1}))f_P^m(\bar{v}) \tag{8.23}$$

- along the bends

$$\boldsymbol{CS}_B^m(\bar{v}) = \begin{cases} (\boldsymbol{CS}^S + \boldsymbol{D}^{xy}(v_{q0}))(1 - f_B^m(\bar{v})) + \boldsymbol{CS}^B f_B^m(\bar{v}) + \boldsymbol{f}_B^d(\bar{v}), \bar{v} \le 0.5 \\ (\boldsymbol{CS}^S + \boldsymbol{D}^{xy}(v_{q1}))(1 - f_B^m(\bar{v})) + \boldsymbol{CS}^B f_B^m(\bar{v}) + \boldsymbol{f}_B^d(\bar{v}), \bar{v} > 0.5 \end{cases} \tag{8.24}$$

$\boldsymbol{D}_q^{xy} = [D_q^x, D_q^y] = [D^x(v_q), D^y(v_q)]$ denotes the location of the detected template centres, which do not necessarily coincide with the origin of the coordinate system that is defined on the corresponding cutting plane. The morphing functions $f_P^m\colon [0,1] \mapsto [0,1]$ and $f_B^m\colon [0,1] \mapsto [0,1]$ between $\boldsymbol{CS}^S$, $\boldsymbol{CS}^P$, and $\boldsymbol{CS}^B$ are modelled by piecewise-linear functions with four and six control points, respectively:

$$\boldsymbol{cp}_P^m = \begin{bmatrix} \bar{v} = 0 & f_P^m(0) = 0 \\ \xi_1 & f_P^m(\xi_1) = 0 \\ \xi_2 & f_P^m(\xi_2) = 1 \\ \bar{v} = 1 & f_P^m(1) = 1 \end{bmatrix} \qquad \boldsymbol{cp}_B^m = \begin{bmatrix} \bar{v} = 0 & f_B^m(0) = 0 \\ \xi_1 & f_B^m(\xi_1) = 0 \\ \xi_2 & f_B^m(\xi_2) = 1 \\ \xi_3 & f_B^m(\xi_3) = 1 \\ \xi_4 & f_B^m(\xi_4) = 0 \\ \bar{v} = 1 & f_B^m(1) = 0 \end{bmatrix}$$

The last operand $\boldsymbol{f}_B^d(\bar{v})$ in equation 8.24 is an offset function that compensates for the fact the 3D alignment is not exactly parallel to the track floor and the sidewalls with

$$\boldsymbol{f}_B^d(\bar{v}) = \begin{bmatrix} f_B^{dx}(\bar{v})\colon [0,1] \mapsto [-w^{track}/2, w^{track}/2] \\ f_B^{dy}(\bar{v})\colon [0,1] \mapsto [-h^{wall}, h^{wall}] \end{bmatrix}$$

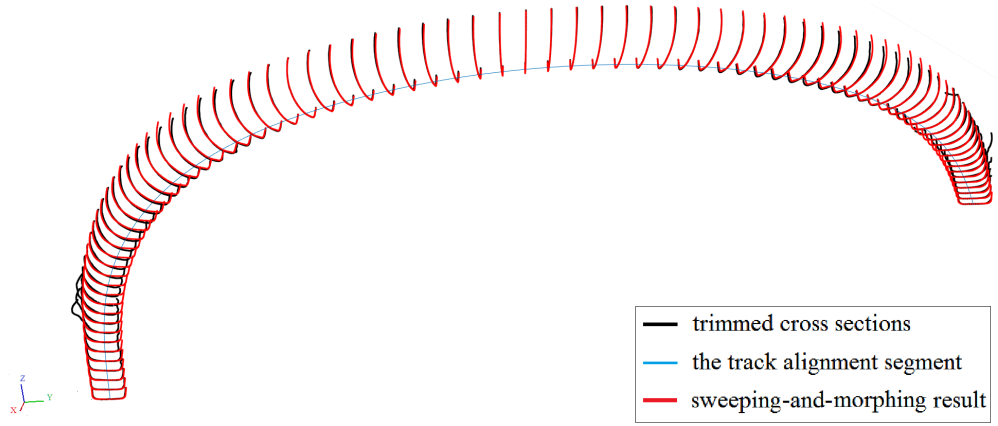
The offsets are modelled by a cubic interpolating spline with endpoint tangents equal to zero (Matlab, 2017b) between five control points:

$$\boldsymbol{cp}_B^{dx} = \begin{bmatrix} \bar{v} = 0 & f_B^{dx}(0) = 0 \\ \bar{v} = 0.25 & f_B^{dx}(0.25) = \psi_1 \\ \bar{v} = 0.5 & f_B^{dx}(0.5) = 0 \\ \bar{v} = 0.75 & f_B^{dx}(0.75) = \psi_2 \\ \bar{v} = 1 & f_B^{dx}(1) = 1 \end{bmatrix} \qquad \boldsymbol{cp}_B^{dy} = \begin{bmatrix} \bar{v} = 0 & f_B^{dy}(0) = 0 \\ \bar{v} = 0.25 & f_B^{dy}(0.25) = \psi_3 \\ \bar{v} = 0.5 & f_B^{dy}(0.5) = 0 \\ \bar{v} = 0.75 & f_B^{dy}(0.75) = \psi_4 \\ \bar{v} = 1 & f_B^{dy}(1) = 1 \end{bmatrix}$$

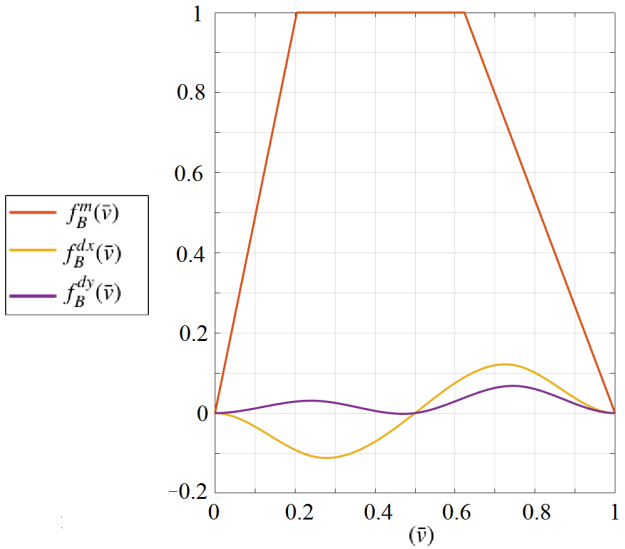
The initial values for ξ^o are acquired by normalising the corresponding segment of the clothoid spline's curvature plot, q.v. figure 8.29, whereas the values for ψ are set to zero. The optimal values for the unknowns $\hat{\xi}$ and $\hat{\psi}$ are acquired by a non-linear optimisation process (Powell, 1978; Matlab, 2017d) based on finite central difference approximation of the derivatives of equation 8.25. The final set of cross-sectional curves is estimated using equations 8.22–8.24 with $\boldsymbol{f}_B^d(\bar{v}) = \boldsymbol{0}$ and $\boldsymbol{D}_q^{xy} = \boldsymbol{0}$. An example of the result on a bend, together with the morphing and offset functions, is shown in figure 8.30.

$$\underset{\boldsymbol{\xi},\boldsymbol{\psi}}{argmin}\left(\sum_{\bar{v}} \sum_i \boldsymbol{d}^2\Big(\boldsymbol{CS}(\bar{v}), spl(\boldsymbol{CS}^m(\bar{v}))\Big) + \sum_{\bar{v}} \sum_i \boldsymbol{d}^2\Big(spl(\boldsymbol{CS}^m(\bar{v})), \boldsymbol{CS}(\bar{v})\Big)\right)\eta(\bar{v}) \qquad (8.25)$$

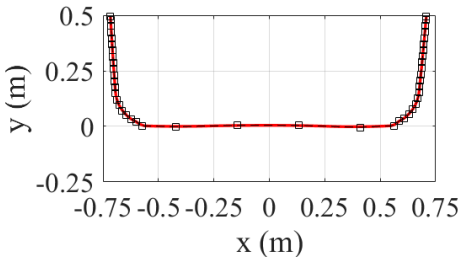
subject to $\xi_2 - \xi_1 > 5(v_{q1} - v_{q0})$ and $\xi_4 - \xi_3 > 5(v_{q1} - v_{q0})$.

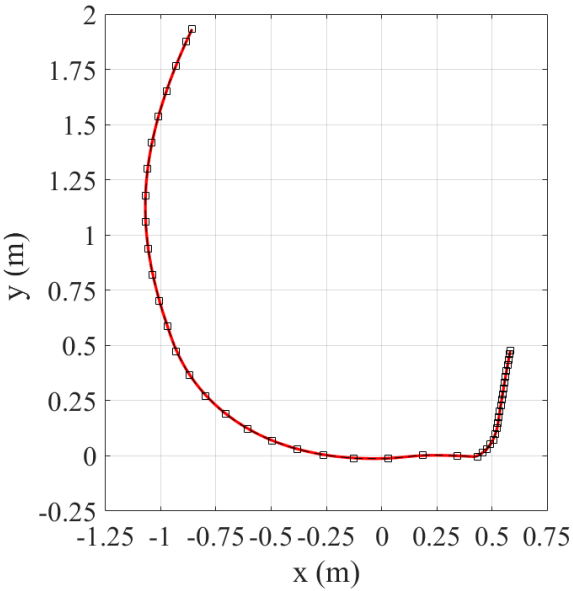


(a) Sweeping-and-morphing.



(b) Morphing and offset functions.





(c) Typical cross-sectional curves for a bend segment, and the corresponding ontrol points, generated from the characteristic cross sections.

Figure 8.30: Sweeping-and-morphing along a bend segment.

Chapter Summary

Three optimisation frameworks were described for the automation of the track modelling process starting from a 3D mesh. The problems that were addressed were the datum definition of a photogrammetric model of a track and the extraction of its alignment and cross sectional curves in order to create a complete 3D model of its surface. For all three cases, the challenge was to generate a reliable approximate solution, from datasets that include outliers, which could then be refined using a standard gradient-based least squares optimisation method. All three of them relied on the geometry of the track's alignment whose initialisation was based on the sensor's trajectory, thus overriding the medial axis detection problem. The clothoid spline fitting and the sweeping-and-morphing algorithm were designed to extract information from the correctly reconstructed parts of the track and propagate them in order to fill in missing information and create a smooth track representation. Their success relies on whether or not the existing information was reliably extracted. For the clothoid spline, the most challenging part was generating the correct number of straight line segments from the detected template centres and combining them to create the control polygon. Compared to the state-of-the-art methods, the proposed algorithm, even though potentially the most computationally demanding—the most computationally intensive part being independently fitting a combined bend to each polygon vertex—was the most successful in reconstructing an alignment that captures the design intent of the track;. The sweeping-and-morphing algorithm will only be successful if a sufficient part of each bend is correctly reconstructed in order to detect a characteristic cross sectional curve towards the mid-arc; an appropriate sampling interval for the cross section extraction will also affect the effectiveness of the detection process. Nevertheless, the boundaries of the track's surface could be inferred, even when dealing with the full 3D point clouds of the site, by using a standard set of cross-sectional template curves and dynamically adjusting them to the measured data. The datum definition, assuming that it could not be directly defined by reference measurements, comprised two processes within a common network. The first one relied on publicly available—potentially unreliable—approximate information about the geometry of the alignment alone, i.e. length, slope, and maximum vertical drop. Once an initial reconstruction of the track's surface is available, its scale could be defined based on the average width of the extracted cross sections. Subsequently, its orientation was optimised in order to minimise the angle of the track's floor on the cross sections with respect to the horizontal plane.

Part IV

Synthesis

This part combines the constituent image-based track scanning and 3D modelling methods which were described in detail in part III. An overview of the complete workflow is given in chapter 9. Details of its implementation on two track sites and further discussion on its performance, limitations, and how it compares with other techniques are presented in chapters 10 and 11.

Chapter 9

The Proposed Workflow

In this chapter, an image-based 3D scanning, reconstruction, and modelling workflow is described, together with recommendations for its implementation. The limiting case scenario of visiting a new track for the first time with only 1–2 hours available on the site is considered. It is assumed that only the least possible amount of prior information about the specific track, q.v. table 2.1, and no reference information, i.e. GCPs or design plans, is available. The aim is to create a 3D model of the track using low-cost, off-the-shelf equipment for data acquisition and a 3D reconstruction and modelling workflow that requires minimal user-interaction. No special treatment for the ice surface is considered, e.g. paint or artificial targets.

9.1 Resources

It is assumed that the entire workload will be carried out by a single person. Use of a full frame, DSLR camera with a 28 mm prime lens is recommended. If possible, the images should be captured in raw format to allow for further processing if necessary—thus, adequate storage space should be available assuming that the size of each image file will be at least 25 megabytes—but images with standard jpeg compression are acceptable, as well. If the track is scanned in the iced condition, the reduced battery life at low temperature should be taken into account; footwear with ice cleats is also necessary.

The software prototype for the track reconstruction was built in (MATLAB, 2017) using the open source geometric processing engine of CloudCompare (2017) and MeshLab (Cignoni et al., 2008) and was based on the photogrammetric engine of the standard version of PhotoScan (2018)—the professional version was used for the assessment of the results. Therefore, their requirements should be taken into account with respect to the characteristics of the computing system;

for the case studies that are presented in this study, a standard laptop was used—see chapter 3.6.1.1 for its technical specifications.

9.2 Data Collection

The implementation of imaging block 5, 11, or 26 at the standing position, or as low as necessary in order for the track edge on the outside of the bends to be visible on the images, is recommended, q.v. figures 7.2 and 7.3a. The images should be captured in an orderly fashion from start to finish, starting 4–6 m before the push-line. The central part of each image should be occupied by the part of the track's surface that lies approximately 10–15 m ahead. The size of the along-track intervals should be defined according to the track length and the available time on site but should not exceed 4–5 m. For the first few meters up to the start line, where the track is wider, more images should be captured at shorter intervals, q.v. figure 9.1, in order to facilitate the automatic start line detection. The horizontal distance of the camera from the sidewalls should be at least 0.35 m to allow coverage of the vertical part of their surface. The *Y* axis of the camera coordinate system, q.v. figure 3.5, of the first image should be approximately vertical; all the other images should be taken in landscape mode, as well, so that features on either side of the track can be tracked over longer image sequences. Roll variations are desirable but images in portrait mode would reduce the footprint of the field of view due to the proximity of the track floor and roof to the camera.

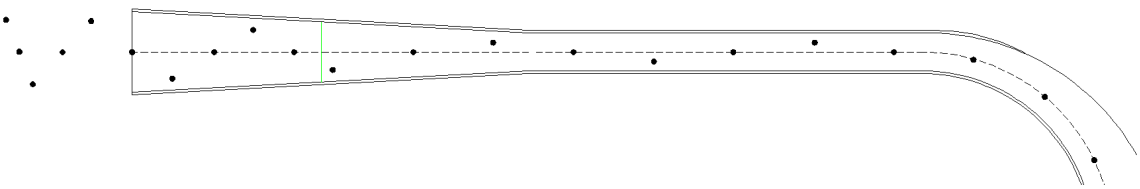


Figure 9.1: A schematic example of the implementation of generic block 26.
— The track's start line.

The width of the push-off and start line, the average track width along the straight segments, and the maximum height of the track's outer edge from the floor should be confirmed by tape measurements on-site. If possible, track scanning should be scheduled for an overcast day to allow for diffuse lighting in order to avoid extreme light variations. The camera should be operated in the aperture priority mode. Most lenses perform best at f/8; adjusting the aperture one or two stops up or down in order to achieve optimal exposure—while maintaining the exposure time and ISO setting close to 1/100 s and 200, respectively—is not expected to significantly affect the geometric

stability of the camera with respect to the accuracy requirements of track scanning. Nonetheless, these settings strongly depend on the conditions at the site on the day. The image stabilisation and sharpening features should disabled and the focus mechanism manually fixed at the lens' hyperfocal distance in order to provide an optimal depth of field for the utilised range of f-stops.

9.3 Data Processing

A common sensor model is assumed for all images captured with the same camera. An initial estimation for the interior orientation parameters should be acquired from an off-site calibration, q.v. appendix D, and include focal length, principal point coordinates, three radial and two tangential distortion coefficients. These values will be refined within a self-calibrating bundle adjustment. If the interior orientation of the camera is unknown, it can be initialised from an on-site self-calibrating bundle adjustment using the first few images of the track that include the start line so that its width can be taken into account in the process. If the track is in the iced state and the ice is glossy, an alternative test-field with adequate depth variations, rich texture, and diffuse reflectance properties should be sought for.

As the number of images that will be collected on site may be well below what is required due to time restrictions, no image is to be excluded from processing due to low quality, at least not until after the initial orientation process. The image orientation process should be conducted using the full image resolution without limiting the number of detected features, unless the required processing time is prohibitive. In order to facilitate the image matching process, approximate exterior orientation parameters should be generated based on the assumed geometry of the generic imaging block, i.e. a forward-facing camera moving on a straight line and capturing images at approximately 2 m long intervals. These coordinates will only be used as reference in order to establish the sequential nature of the image acquisition process and not constrain the exterior orientation parameters during the bundle adjustments.

After the incremental, local, free self-calibrating bundle adjustments for the generation of approximate values for the entire parameter set, a global self-calibrating bundle adjustment—the SfM solution—should be performed iteratively using conservative data pruning thresholds; i.e. maximum relative reprojection error, reconstruction uncertainty, and projection accuracy set to 1 pixels—at the corresponding image scale—500, and 30, respectively, until the total RMS reprojection error is below 1 pixel. Stricter threshold values can be used on an ad hoc basis as long as a sufficient number of feature matches between successive images, which can yield the reconstruction of the entire image set along the track in a single network, is maintained. PhotoScan's

default uncertainty estimations for the approximate camera coordinates, camera rotation angles, and feature point coordinates of 10 m, 2° and 1 pixel, respectively, were deemed appropriate for the bundle adjustments.

The parameter settings for the 3D dense point cloud and mesh generation depend on the available processing capabilities, as there is no straightforward way of directly controlling the number of 3D points and triangle faces. The workload is expected to involve on average 500–2000 images with a resolution of 10–30 MP; therefore, reducing the image resolution to 1.5–6 MP and applying the strictest filtering option is suggested for dense image matching. The 3D mesh should be reduced in order to comprise less than 1 face for every 5–15 point cloud points. Segmentation of the track on the images is essential to the start line detection process but the detection of the track's surface edges cannot be guaranteed and needs to be taken into account during the 3D geometry processing stage, regardless of whether or not masked images were used during dense matching.

The implementation details of the algorithms that are used throughout the process and the related literature were described in the corresponding sections in part II, chapters 7 and 8. The 3D reconstruction process is based on the standard PhotoScan processing pipeline, q.v. section 3.6.1.1. An overview of the 3D modelling pipeline is given in figure 9.2. Recommendations for the values for the used-defined parameters that are required as input for and modelling process are provided in the following chapters.

Chapter Summary

An overview of the proposed workflow, which is illustrated in figure 9.2, was presented in this chapter. It provided guidelines with respect to the required resources and the implementation process, which were detailed in the previous chapters. The described workflow was used as the basis for carrying out the case studies that are described in chapters 10 and 11, in order to assess its effectiveness and efficiency.

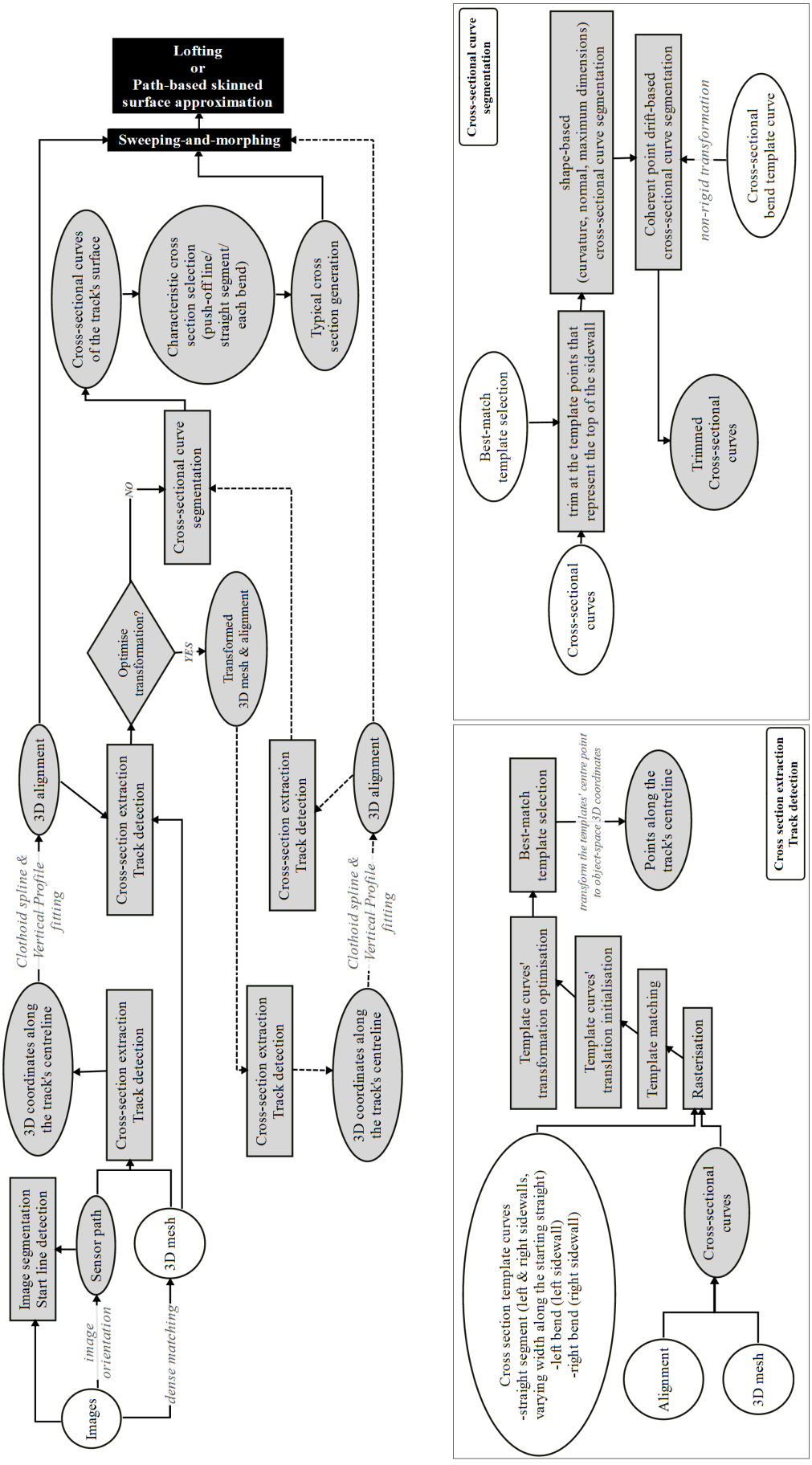


Figure 9.2: Track modelling workflow.

Chapter 10

Lillehammer Track Scanning

A trial was conducted in Lillehammer, q.v. table2.1 and figure 1.1, in order to assess whether or not the entire length of a track in the iced condition could be scanned using the image-based scanning technique that was described in chapter 9. The hypothesis was that implementing the imaging configuration that appeared to be the most reliable for sparse reconstruction during the network design simulations—namely, collecting stereo pairs according to block 11 at regular intervals along a smooth trajectory down the track, q.v. section 7.1—using a non-metric, but precalibrated, high resolution DSLR camera with a prime lens, would overcome the problems that were encountered during the Winterberg track scanning trial, q.v. section 5.2, and allow for the estimation of the relative orientation of the entire image sequence. Section 10.1 presents the data collection and processing details. The shortcomings of the image acquisition process and potential solutions are discussed in section 10.2. The track alignment and surface reconstruction results are compared with those that were generated with alternative methods in sections 10.3 and 10.4, respectively.

10.1 Image-Based Scanning and Sparse Reconstruction

A Canon EOS5D Mark II DSLR camera with a Canon EF 28 mm f/1.8 USM prime lens was used on an overcast October day with the temperature at $-10^{\circ}C$. The concrete track was covered by a shiny layer of ice and the surrounding environment was covered in snow. Approximately 90% of the track's length was roofed and the blinds were limiting the view to the surrounding environment for almost 50% of it. During a track walk along an approximately 1450 m long segment of the track, from behind the push-off stretch until a few metres after the finish line, 755 still images with a 21 MP resolution (5616×3744 pixels) were captured—this dataset will be called $Lil21mp$ hereafter. All images were saved in raw format and converted to the tagged image file format



Figure 10.1: Views of the Lillehammer track on the day it was scanned.

(TIFF). Block 11 at the standing position—q.v. figure 7.2 and 7.3a—was used to capture stereo pairs, except for the first 50 m along the starting stretch where image triplets were acquired as the track is wider. The camera was operated in the aperture priority mode with the f-number set to 8. The image stabilisation and sharpening features were disabled and the focus mechanism was manually fixed at 3.3 m, providing a depth of field from approximately 1.6 m from the camera to infinity. The plan was to maintain the camera station intervals slightly below 4 m, so that the image acquisition could be completed within the allocated time on the on that day, which was 90 minutes; the average scanning pace was 7.2 sec/image or 15.8 m/min. A 20 minutes long video stream with a resolution of 1920×1080 pixels was also acquired the following day, under similar conditions by a forward-facing camera moving along the track's centreline.

10.1.1 Reference measurements

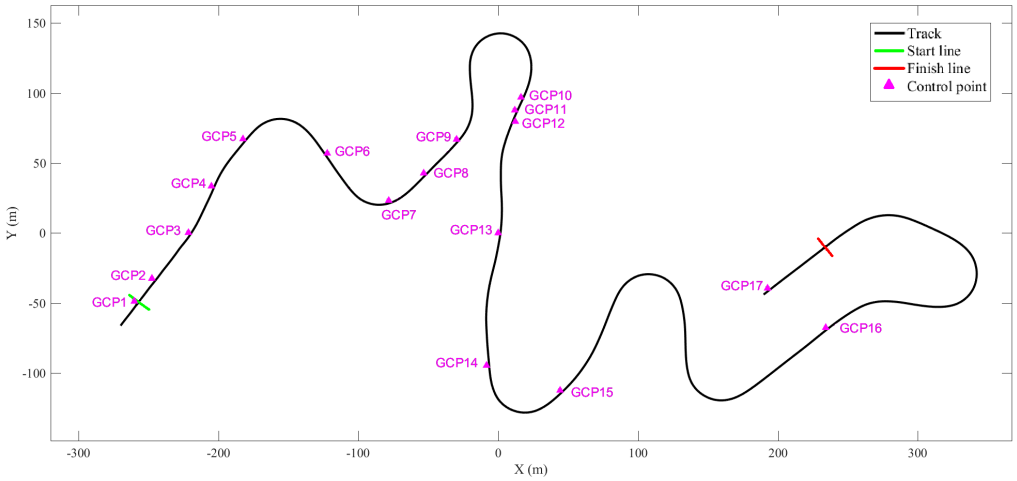
In order to be able to assess the quality of the photogrammetric results at discrete positions along the track, the coordinates of non-signalised points—which were visible and identifiable on the images, q.v. figure 10.2—were measured around the track by means of a GNSS survey using the single baseline real time kinematic (RTK) technique, q.v. table E.1. Selecting the objects that could serve as targets was challenging as most of the permanent objects outside the track were not visible from inside the track and could barely accommodate the GNSS antenna on a tribrach, q.v. figure 10.2b. Some distinctive objects were either inappropriate for the antenna to be placed on or flat horizontal surfaces at the same level as the camera trajectory which could make their identification and measurement on the images nearly impossible. Considering the accuracy potential of the applied surveying technique and the results that were described in section 2.3 and chapter 5, a relative positional accuracy of 5 cm for the ground control point coordinates was deemed adequate for this project and—taking into account the unavoidable antenna centring errors—achievable by the GNSS RTK survey.



(a) Ground control point 13.



(b) Ground control point 16.



(c) Plan view of the track with the GCP locations.

Figure 10.2: Ground control points at the Lillehammer track.

Table 10.1: Parameter settings for the image orientation process of the Lillehammer dataset within PhotoScan.

Uncertainty Estimations			
camera position	camera rotation	GCP coordinates	image coordinates
10 m	2°	0.05 m	1 pixel

Processing Parameter Settings				
Resolution	Tie Point Limit	Match Point Limit	Adaptive Camera Model	Pair Preselection
full	no limit	<4000	No	Yes
×4	150000	<4000	No	Yes

10.1.2 Processing protocol

An overview of the image orientation process, following the guidelines that were given in section 9.3, is shown in figure 10.3. The camera's interior orientation was initialised with an estimation from an off-site self-calibrating bundle adjustment using a 3D frame with coded targets, q.v. table D.1. Firstly, the SfM solution was estimated for both datasets. The specific thresholds for the RMS and maximum reprojection error, projection accuracy, and reconstruction uncertainty for each one of the bundle adjustments that were calculated for this dataset are reported in table 10.2. For the constrained self-calibrating adjustments that were conducted in order to generate reference datasets, the non-signalised ground control points were manually measured on the images with at least three image projections observed for each one. The a priori uncertainty estimations that were used in the bundle adjustments are summarised in table 10.1. A common iterative scheme was used until the total reprojection and control point coordinate RMSE were at most 1 pixel and 5 cm, respectively, or no significant change was evident after an iteration.

10.1.3 Image Orientation

Despite several attempts to estimate the relative orientation of all the images in the $Lil21mp$ set within a single network, using all possible parameter setting combinations within PhotoScan—with each attempt lasting between 2–11 hours—this was not possible; the causes are investigated further in section 10.2. Thus, an initial assessment of the reconstruction accuracy and repeatability of the image-based technique was only conducted for the first 350 m of the track. In order to estimate the overall quality of the track's shape, 5 SfM solutions were estimated, q.v. table 10.2. Each sparse reconstruction was generated from an independent feature detection and matching process using the same 243 images and parameter settings. The total RMSE of a similarity transformation that was calculated for each one, based on GCPs 1–9, was below 10 cm for 4 of the trials—namely A, B, D, E. However, for trial C it was almost 20 times worse, even though its internal accuracy appeared to be as good as the other 4. The residuals and total RMSE of the similarity transformation and a constrained bundle adjustment using 4 GCPs at 75–150 m intervals for trial B, are shown in figure 10.4. These sparse reconstructions will hereafter be referred to as $Lil350B_{sim}$ and $Lil350B_{cba}$, respectively.

Two attempts to reconstruct the entire length of the track using frames extracted from the recorded video stream at 1 and 2 fps were also unsuccessful. Subsequently, a set of 6000 frames was extracted at 5 fps corresponding to along-track intervals of approximately 0.25 m. As the image redundancy was critical in facilitating the matching process, q.v. chapter 5, only 130 frames with

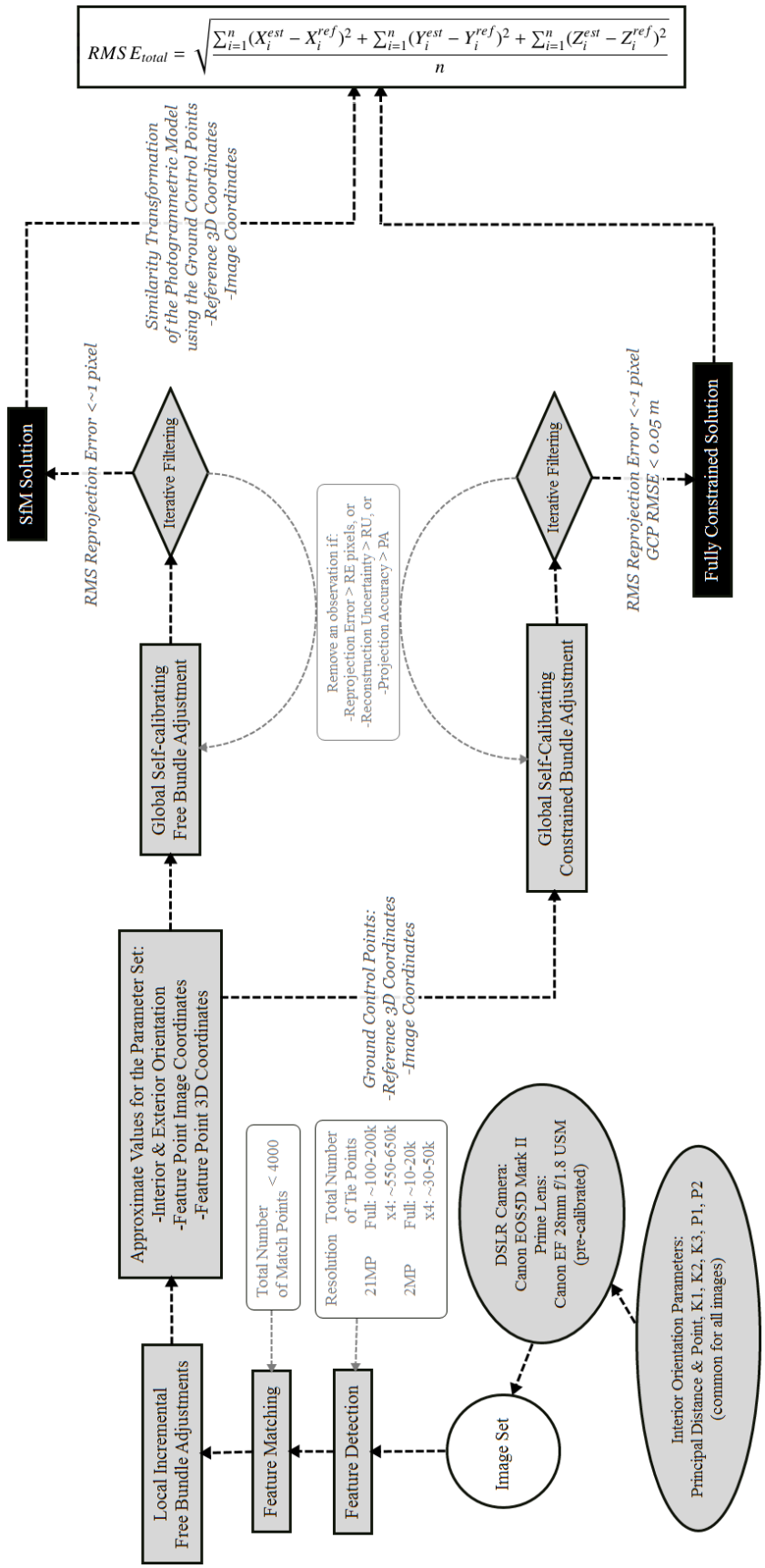


Figure 10.3: The processing protocol for Image orientation within PhotoScan. The ground control point coordinates are introduced after the approximate value generation in order to be able to assess the integrity of the SfM solution.

Table 10.2: The results of five trials for the relative orientation and similarity transformation of the high resolution dataset *Lil*350*B*, using three filtering schemes.

Trial ID	A	B	C	D	E
RE:- RU:- PA:-					
Points	170342	170929	169127	170116	170728
GCP RMSE	0.069	0.051[b]	2.022	0.091	0.063
Total RMS reprojection error	0.60	0.61	0.60	0.60	0.60
max reprojection error	0.40	0.40	0.46	0.42	0.41
RE:0.20 RU:300 PA:15					
Points	127762	125845	126468	128062	126500
GCP RMSE	0.116	0.068	1.950	0.175	0.117
Total RMS reprojection error	0.35	0.33	0.34	0.35	0.34
RE:0.10 RU:150 PA:10					
Points	69218	70079	67658	69637	68538
GCP RMSE	0.091	0.171	1.927	0.109	0.068
Total RMS reprojection error	0.18	0.18	0.18	0.18	0.18

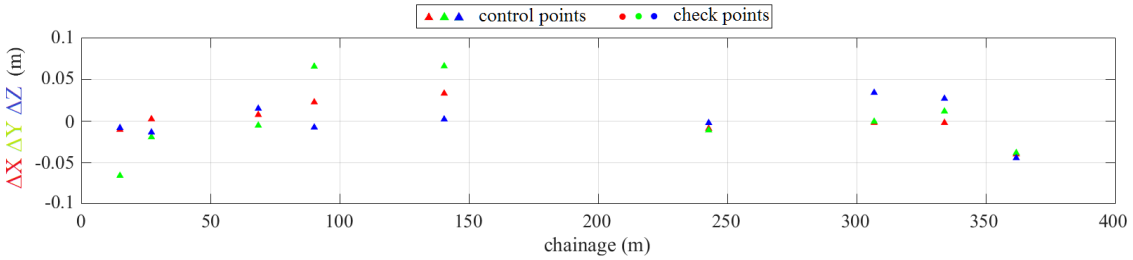
[a] RE:max Reprojection Error (pixels)
RU: max Reconstruction Uncertainty
PA: max Projection Accuracy

[b] $Lil350B_{sim}$: this imaging network is used in section 10.3

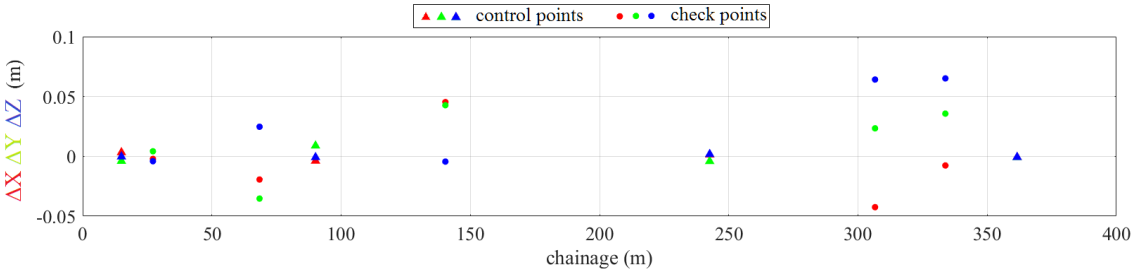
a quality metric lower than 0.45 were excluded from further processing as they were obviously blurred. Even though the correspondence for a sufficient number of feature points between every successive image pair was correctly established, after a 3 hour long matching process, the relative orientation of the entire sequence within a single automatic step—which lasted another 3 hours—was not successful. This was only possible after manually selecting successive image subsets, each one corresponding to approximately 50–100 m of track length, in order to re-estimate the triangulation for the corresponding feature matches. The total RMSE of the similarity transformation of the SfM solution, which was calculated using all the ground control points, was 20.86 m, q.v. figure 10.5a. A significantly improved result was obtained by using all the available GCPs in order to generate reliable approximate parameter values for a constrained self-calibrating bundle adjustment, as the total RMSE at the 3 control and 14 check points was reduced to 0.01 m and 0.43 m, respectively—this solution will be referred to as $Vid5fps_{cba}$—q.v. figure 10.6.

10.2 Online Feedback

Having a clear plan for image acquisition, as a result of the network design process, can increase the probability of a successful track reconstruction; however, several aspects of the task can be



(a) Residuals of the similarity transformation of $Lil350B_{sim}$ using all the available GCPs.



(b) Residuals of a fully constrained bundle adjustment ($Lil350B_{cba}$) using GCPs 1, 4, 6, and 9 as control points.

Figure 10.4: The control and check point coordinate residuals for the similarity transformation and constrained bundle adjustment of dataset *Lil*350*B*.

unpredictable. For example, glossy ice or extreme lighting variations within the depth of field may require shorter along-track intervals between the camera stations; conversely, an unanticipated event may reduce the available time on the track and call for longer intervals which may prove to be too long and prevent the relative orientation of the entire image set within a single network.

Following the failed attempts to scan the entire length of the Lillehammer and Winterberg tracks, the requirements for an online feedback mechanism were considered. It is quite clear that the problem arises when not enough feature matches are generated between two or more images with an overlapping field of view, causing the sequence to break. However, in an ideal solution, the user should be able to prevent this from happening without having to actually run the image detection and matching process. The hypothesis was that an estimation of the image quality could provide a clear indication to this end. It was tested by retrospectively assessing whether or not an estimation of the image quality at the time of acquisition would be informative to the user, for images that were proved to be problematic during post-processing.

One of the observations that was made during the preliminary trials in chapter 5, was that the quality metric used in (PhotoScan, 2018) was not always representative of the image quality. Thus, four additional quality measures were estimated for the image sets that were collected along the Winterberg and the Lillehammer track, both of them comprising stereo pairs of average quality



(a) Similarity transformation of the SfM solution using all the available GCPs ($Vid5fps_{sim}$).

Total RMSE at the control points: 20.86 m.



(b) Constrained bundle adjustment using control points 1, 14, and 17 ($Vid5fps_{cba}$); q.v. figure 10.6 .

Total RMSE at the control points: 0.01 m.

Total RMSE at check points: 0.43 m.

Figure 10.5: Sparse reconstruction of the Lillehammer track from the $Vid5fps$ dataset. Notice the gradual scale drift when no ground control is used in the bundle adjustment.

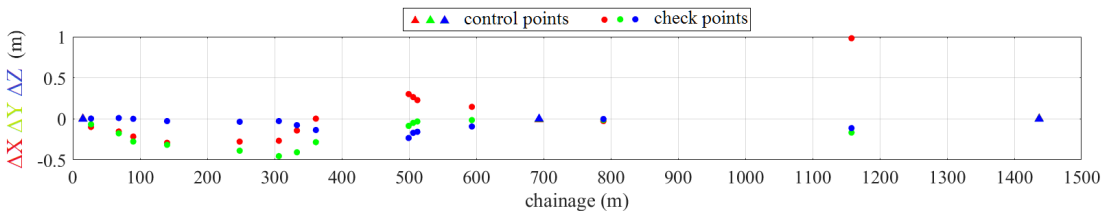


Figure 10.6: The point coordinate residuals of $Vid5fps_{cba}$ after a fully constrained bundle adjustment using GCPs 1, 14, and 17 as control points.

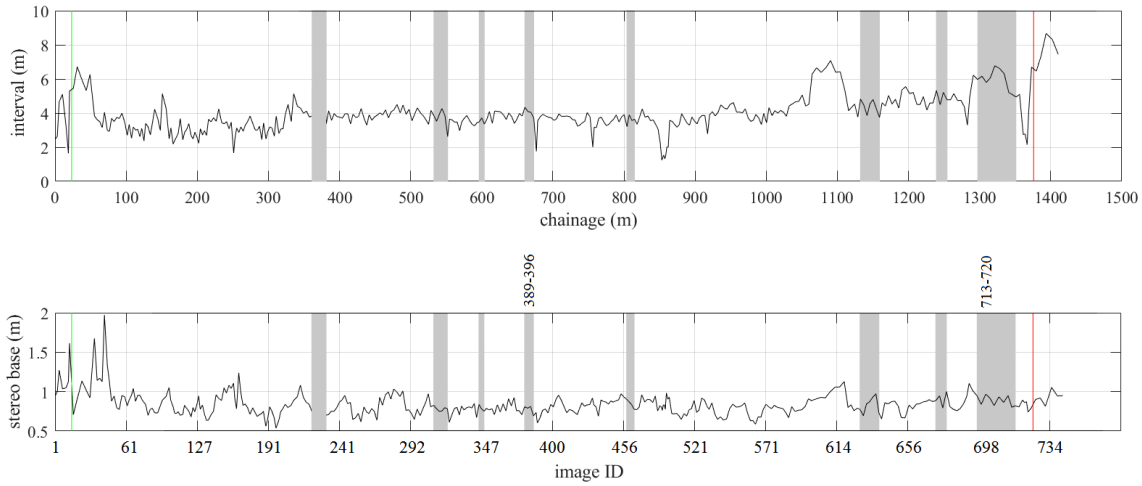


Figure 10.7: The along-track distances between the successive camera stations and the stereo base for each image pair at the Lillehammer track. The grey areas indicate track segments along which the orientation of still frames was problematic, q.v. section 10.2. Note that these values were estimated from a successfully oriented dataset than included both video frames and still images, hence the gap at chainage 0+375 m.

collected with a hand-held camera. In both cases, the exposure time for each image was the longest possible for hand-held photography; therefore, motion blur was to be expected in all frames. For the former, the camera was operated in automatic mode; for the latter it was set to aperture priority with a constant aperture size. In figures C.1 and C.2, PhotoScan's quality metric, the blur metric, the variance of the Laplacian, as well as, the BRISQUE and NIQE metrics, q.v. section 3.2.1, are plotted for each image sequence, together with the corresponding camera settings, i.e. exposure time, f-number, and ISO. Higher values indicate better quality and the areas highlighted in gray correspond to images whose alignment was problematic. The along-track distances between the successive camera stations and the stereo base for each image pair are shown in figure 5.5 and 10.7. [1]

As no distinctive pattern could be identified in the plots it was hypothesized, upon closer inspection of the problematic sequences, that the major factors causing the alignment failure was the large intervals between the camera stations in combination with the poor reflectance properties of the scene itself, which could not be conveyed by the aforementioned metrics. A total of 8 image sequences from the Winterberg and Lilehammer track were considered, q.v. table 10.3 in order to identify potential correlations between the image orientation success rate and the aforementioned

[1] The increase towards the end of the Lillehammer track coincided with a public announcement calling for everyone to vacate the track within 15 minutes for the oncoming sliders

factors: 5 problematic—i.e. 1, 3, 4, 6, 8—plus 3 more of poor quality with various reflectance properties, which were successfully aligned and were used as reference, q.v. appendix C. Each sequence comprised 4 stereo pairs; the reflectance properties of the depicted part of the scene was assigned a quality level from very poor to very good, based on visual inspection, aiming to describe the extent of specular highlights on the surface of the ice and the surrounding environment, which could affect the image matching process.

Table 10.3: Summary of the geometric properties and reflectance characteristics—based on visual inspection—of 8 image sequences comprising 4 stereo pairs collected along the Wintergberg and Lillehammer track, and their effect on the image orientation success, q.v. tables C.1–C.8.

Sequence ID	Track Name	Image Set	Average Quality	Reflectance	Min Base (m)	Max Interval (m)	Image Orientation Success Rate		
							Resolution		
							1/16	1/4	full
1	Winterberg	191–198	0.70	Poor	0.55	6.5	0%	20%	66%
2		201–208	0.59	Average	0.55	4.0	96%	94%	100%
3		239–246	0.68	Poor	0.55	6.5	0%	0%	0%
4		501–508	0.83	Average	0.55	7.5	56%	100%	4%
5	Lillehammer	105–112	0.62	Poor	0.70	2.8	42%	54%	74%
6		389–396	0.70	Very Poor	0.70	4.3	0%	0%	0%
7		581–588	0.58	Good	0.60	3.9	100%	100%	100%
8		713–720	0.52	Poor	0.80	6.1	2%	72%	58%

Table 10.3 summarises the results from 50 image orientation attempts at three processing levels—using the images at full, 1/4, and 1/16 of their resolution. It can be said that the lowest success rates correspond to image sequences along surfaces of poor reflectance qualities, i.e. 1, 3, 6. One way to alleviate the negative effect was by shorter intervals, i.e. sequences 5 and 7 but the larger intervals was not a problem for sequence 4, with the higher average quality metric between them. Nevertheless, there exist image sequences among the entire dataset with lower quality and acquired at comparable intervals which were successfully aligned. The apparent discrepancy at the success rates between the low and the higher resolutions for sequences 1, 4, and 8 is an indication of the fact that the decision to reject an image on the basis of the processing result using a downsampled version, may not be correct.

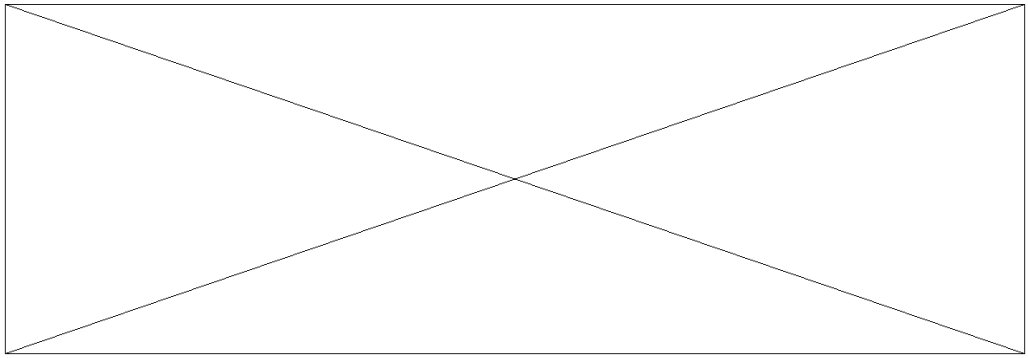


Figure 10.8: Plan view and vertical profile along the Lillehammer track floor, adapted from Zhang et al. (1995, p. 12,fig. 4). Copyrighted material removed, see figure F.1.

10.3 Comparisons with Alternative Data Sources

In some cases, the as-designed typical cross sections of the track may be available or the general shape of some characteristic cross section may be generated by tape measurements on site, q.v. section 2.2.2. However, acquiring a reliable estimation of the track's longitudinal shape is not always possible or straightforward. This section presents a comparison between four potential data sources for the generation of the 3D geometry of the track's centreline alignment: a reproduction of the design plans, an orthoimage of the site, the photogrammetric survey that was described in section 10.1—namely, $VId5fps_{cba}$ and $Lil350B_{sim}$—and a hand-held laser scanning survey using the Zeb1, q.v. section 3.6.2.

10.3.1 Design Plans

The horizontal alignment and vertical profile which was used in Zhang et al. (1995) for bobsleigh steering simulations was the closest dataset to the actual design plans that could be retrieved for the track. No further information was provided therein about the source or the accuracy of the design plans. The horizontal and vertical plane curves were converted from raster to vector format using a second degree polynomial transformation based on the depicted grid points, q.v. figure 10.8, and were combined in order to create a space curve. The total RMSE of the transformation for the alignment and profile were 0.55 m and 1.45 m, respectively, q.v. figure F.1.

10.3.2 Orthoimagery

A plane curve was extracted from a topographic map and the elevations along its length were sampled from a digital terrain model—both of them were acquired from the online service offered

by the Norwegian Mapping Authority ©Kartverket (2017)—in order to generate a representation of the 3D centreline alignment of the track. Absolute georeferencing of the alignment was not of interest, so the 2D coordinates were transformed from the EUref89 UTM-zone 33 projection to a local coordinate system. As the centreline was traced over the track's roof and the elevations refer to the terrain around the track, it was assumed that the alignment's accuracy was approximately 1.5 m horizontally and 3 m vertically.

10.3.3 Hand-Held Laser Scanning

A laser scanning survey was conducted using the Zeb1 hand-held laser scanner. In order to find a compromise between an acceptable walking pace and overall scan duration and also quantify the unavoidable solution drift for such—longer than recommended—trajectories, four datasets were acquired, q.v. table 10.4. No artificial targets were used, so the datum for each point cloud was defined by the sensor's IMU and range measurements.

- dataset $Zeb1s1$

  The first loop—represented by a green line—started near the upper start area, reached the lower start area at the entrance of bend 4, and returned to the top.

- dataset $Zeb1s2$

  The second loop—represented by a red line—started near the upper start, reached the exit of bend 10, and returned to the top.

- dataset $Zeb1s3$

  The third loop—represented by a cyan line—started at the entrance of bend 10, reached the finish line, and returned; it has an effective along-track overlap of approximately 80 m with $Zeb1s2$.

- dataset $Zeb1s4$

  The fourth loop covered the full length of the track and is represented by the combination of the red and cyan lines.

- dataset $Zeb1s5$

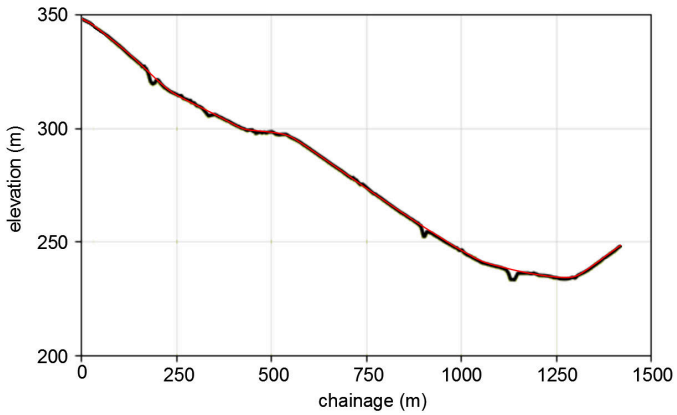
  The previous dataset was processed once more without closing the loop—the continuous red and cyan lines only—in order to have a better indication of the sensor's position estimation drift along the entire track length.



(a) Topographic map.



(b) Orthoimage.



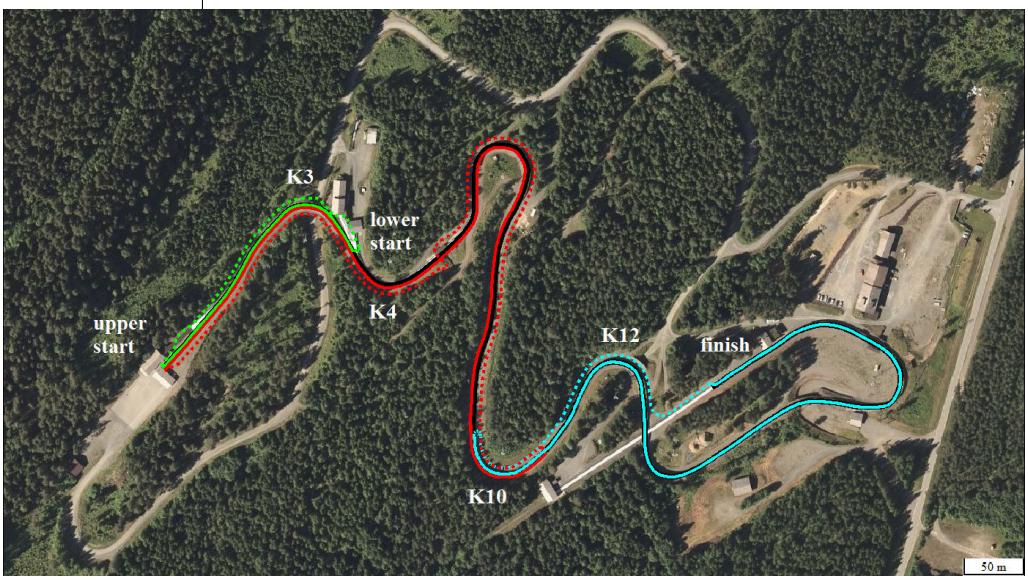
(c) Profile. A smooth curve was traced over the terrain profile so that overpasses, q.v. figures 10.13d and 10.14d, were not taken into account.

Figure 10.9: Lillehammer track topography and along track terrain profile, reproduced from the Norwegian Mapping Authority ©Kartverket (2017, CC BY 4.0).

–·– Horizontal alignment — Terrain profile — Vertical alignment

Table 10.4: Length and duration for each laser-scan survey with the Zeb1 at the Lillehammer track. The on- and off-track sensor trajectories are symbolised with continuous and dashed lines, respectively

Dataset	Zeb1s1	Zeb1s2	Zeb1s3	Zeb1s4	Zeb1s5
Colour code	green	red	cyan	red cyan	red cyan
Line type	continuous dashed	continuous dashed	continuous dashed	continuous dashed	continuous
Total length (m)	500	1600	1040	2560	1420
On-track length (m)	250	780	720	1420	1420
Total duration (min)	13	30	20	50	25
Average pace (m/h)	2307	3200	3120	3072	3408



orthophoto reproduced from the Norwegian Mapping Authority ©Kartverket (2017, CC BY 4.0)

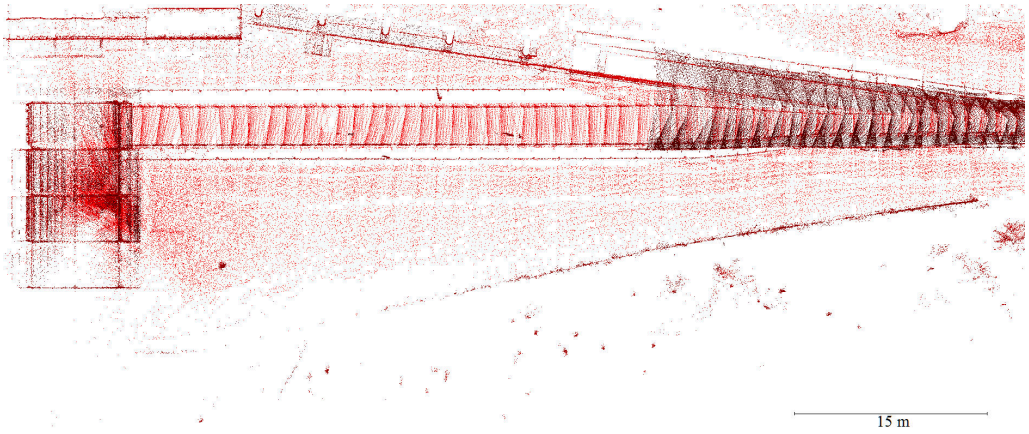
The automatic post-processing of the measurements, using the online web-service that comes with the device, was unsuccessful for datasets $Zeb1s2$ and $Zeb1s4$, and required manual intervention in order to force a loop closure, q.v. section 3.6.2. The start and finish measurements were successfully registered together for the latter; however, this was not possible for the former, presumably due to insufficient overlap at the upper start area between the point clouds collected during departure and arrival, even though enough 3D features correspondences between the point clouds are visible in figure 10.10.

10.3.4 Quality Assessment

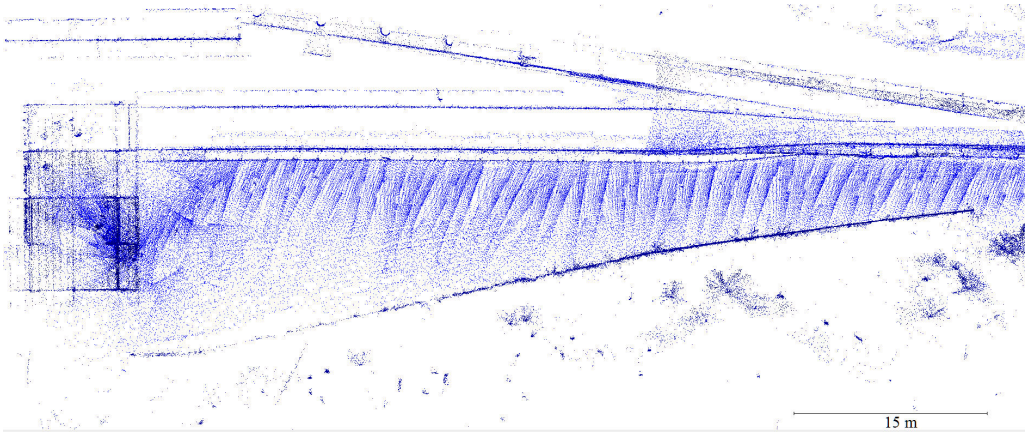
From the design plans and the topographic map, the alignment was directly represented and roughly approximated, respectively. For the photogrammetric and laser scanning surveys, the alignment was represented as a cubic smoothing spline fitted to the sensor's trajectory, q.v. section 8.1.1. It was assumed that all trajectories were parallel to the track's floor; considering the expected accuracy, the approximation error was assumed to be negligible and, in any case, comparable between the datasets under assessment.

A reference alignment was generated from the $Vid5fps_{cba}$ dataset, q.v. section 10.1. Each curve was transformed to the coordinate system of the reference curve by manually initialising its 3D rotation and translation and refining it using the iterative closest point (ICP) algorithm (Besl et al., 1992), q.v. figure 10.11. The external accuracy of each curve, q.v. table 10.5, was estimated at 1 m intervals with respect to the minimum distance to the reference curve, q.v. figure 10.11a. As the reference alignment appeared to be distorted along the last 300 m—where the GCPs were sparsely distributed, q.v. figures 10.2 and 10.11—the assessment was focused on the first 800 m of the track's length, q.v. figure 10.12.

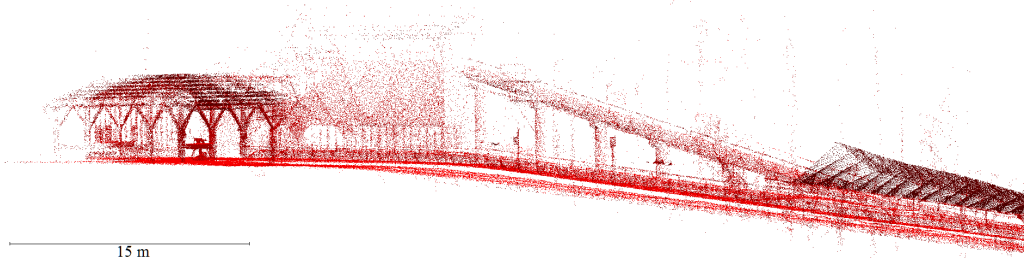
The quality of the as-designed alignment was slightly worse than expected but could be justified by the very small scale of the input raster image; the kink at approximately 100 m from the start is inconsistent with all the other datasets but could be explained by the raster-to-vector transformation residuals of the input, q.v. figure F.1. The alignment that was extracted from the orthoimagery was found to be significantly better than those from $Zeb1s2$, $Zeb1s4$, and $Zeb1s5$ with scan times or lengths that exceeded 25 min and 1400 m, respectively. If the approximation errors of the sensor trajectory-based alignment generation for $Zeb1s1$ and $Lil350B_{sim}$ are ignored—and considering the fact that they were shorter than the rest, which means that a better fit to the reference was achievable—it can be said that a comparable curve could be extracted remotely from the orthoimagery.



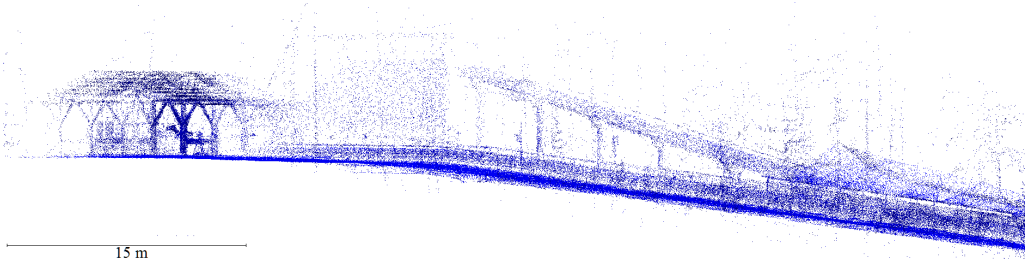
(a) Top view of the upper start area point cloud that was collected during the departure.



(b) Top view of the upper start area point cloud that was collected during the arrival.



(c) Side view of the upper start area point cloud that was collected during the departure.



(d) Side view of the upper start area point cloud that was collected during the arrival.

Figure 10.10: The point clouds that were collected at the starting area of the Lillehammer track at the beginning (red) and finish (blue) of second Zeb1 survey ($Zeb1s2$). Even though common geometric features can be identified, the overlap was not sufficient to automatically close the loop.

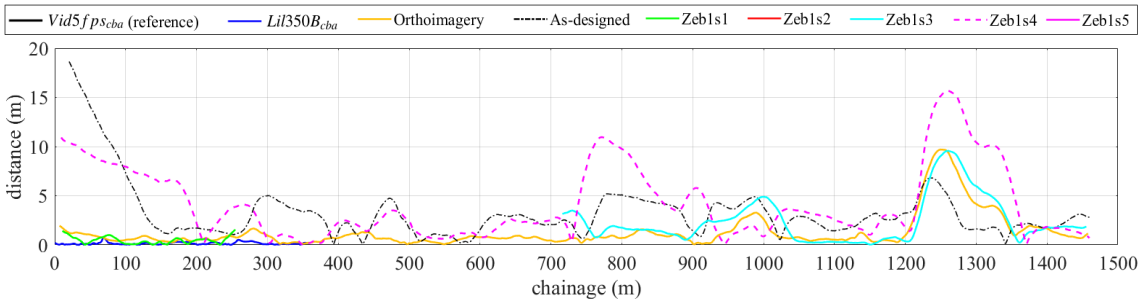
Table 10.5: The along-track quadratic average of the orthogonal distances from the assessed space curves to a reference centreline alignment of the Lillehammer track.

	total RMSE (m)							
				Zeb1 dataset				
	$Lil350B_{sim}$	As-designed	Orthoimagery	1	2	3	4	5
full length	0.20	4.29	1.92	0.59	14.52	3.09	5.14	18.82
first 800 m	0.20	4.77	0.73	0.59	15.47	-	4.60	14.28

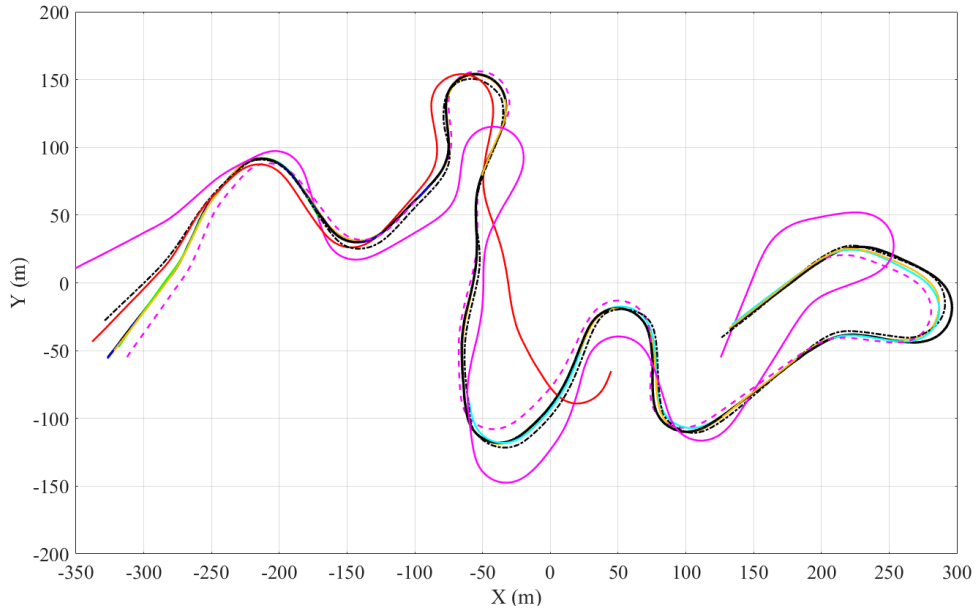
In order to assess the internal accuracy of the $Zeb1$ datasets that closely matched the reference curve, namely $Zeb1s1$ and $Zeb1s3$, two 0.25 m thick point cloud slices were extracted, perpendicular to the alignments, towards the centre of bends 3 and 12, respectively, q.v. figures 10.13a and 10.14a. The blue points on the left side of both slices were collected on the outer surface of the track while walking back to the scan's start location, in order to close the loop from the shortest possible path outside the track. The blue points on the left side of image 10.14b were collected approximately 3 minutes after the rest; the distance along the sensor's trajectory for the two point subsets was 130 m. The blue points on the left side of image 10.13b were collected approximately 7 minutes after the rest; their distance along the sensor's trajectory was 380 m. In theory, a cross section depicting the boundary of the track's wall, which is approximately 0.25–0.35 m thick, should be discernible. The discrepancy was approximately 2 m for $Zeb1s1$ and -0.30 m for $Zeb1s3$.

10.4 Track Modelling

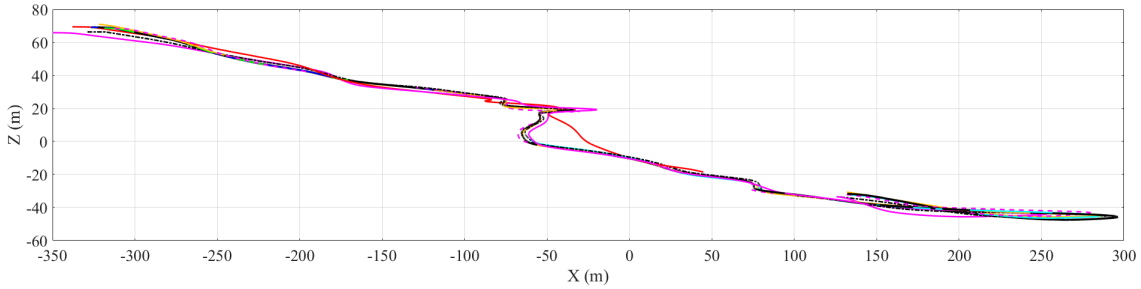
An initial assessment of the proposed 3D modelling method for the un-assisted track surface approximation was conducted based on two 3D meshes that were generated from the $Lil350B_{cba}$ and $LilZeb1s4$ points clouds in PhotoScan and ©Geomagic Studio (2012), respectively. Figures 10.15b and 10.16b show the positions along the track where a template was successfully matched on the extracted cross sections. For $Lil350B_{cba}$, the percentage of the detected left, right, and full templates—with respect to the total number of cross sections, which were extracted at 1 m intervals—was 20%, 22%, and 30%, respectively; 72% in total. For $LilZeb1s4$, a template was detected in 98% of the cross sections—47%, 31%, and 20%, respectively. The difference in the total success rate of the detection process was due to the fact the Zeb1 point clouds—even though potentially sparser, with respect to point density, and of lower range, with respect to the sensor's trajectory—are less noisy and provide higher track coverage.



(a) The distances to the reference alignment with respect to the along-track length. Zeb1s2 and Zeb1s5 are omitted.

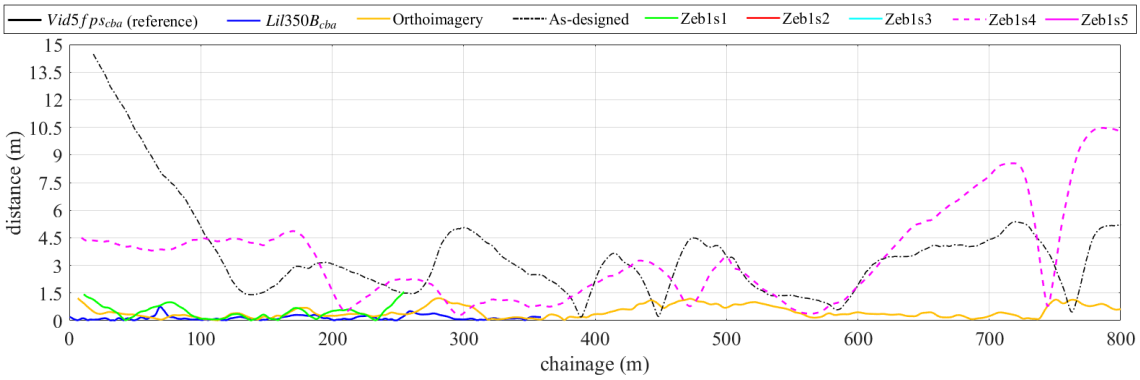


(b) Plan view.

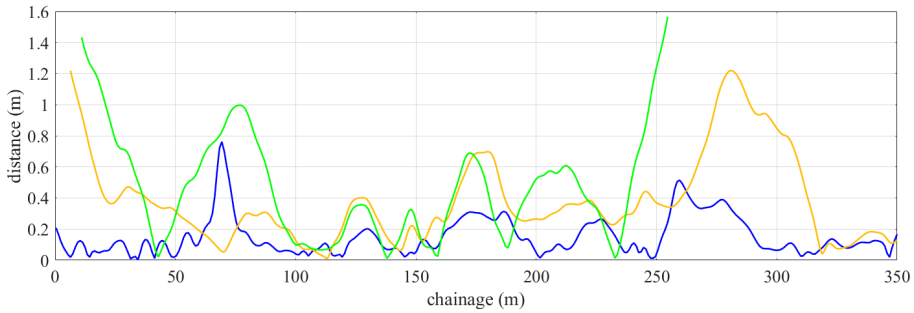


(c) Side view.

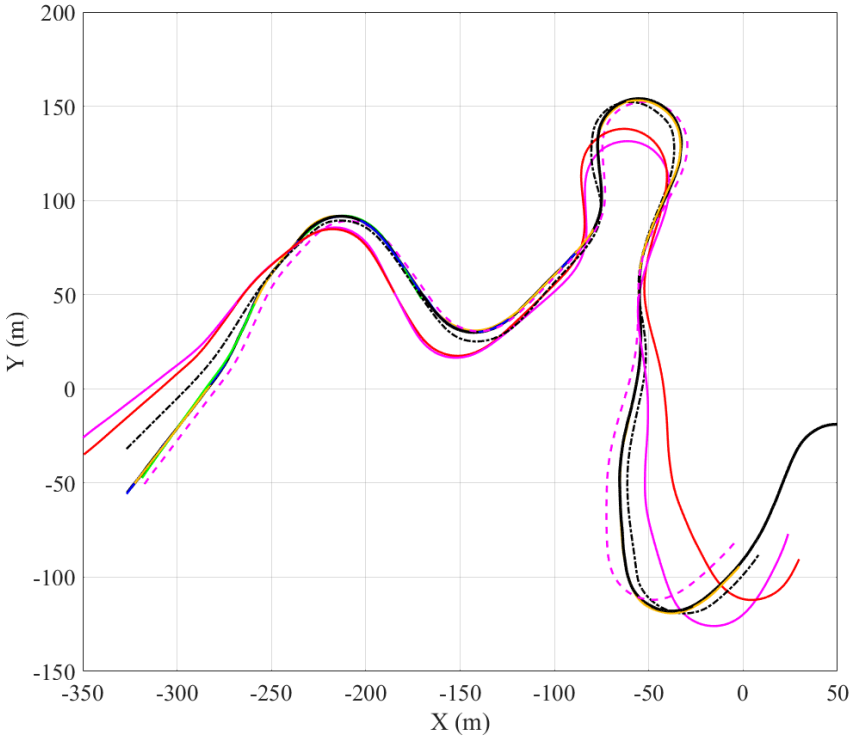
Figure 10.11: Comparisons between the Lillehammer track alignments that were created from various sources (full length).



(a) The distances to the reference alignment with respect to the along-track length. Zeb1s2 and Zeb1s5 are omitted.

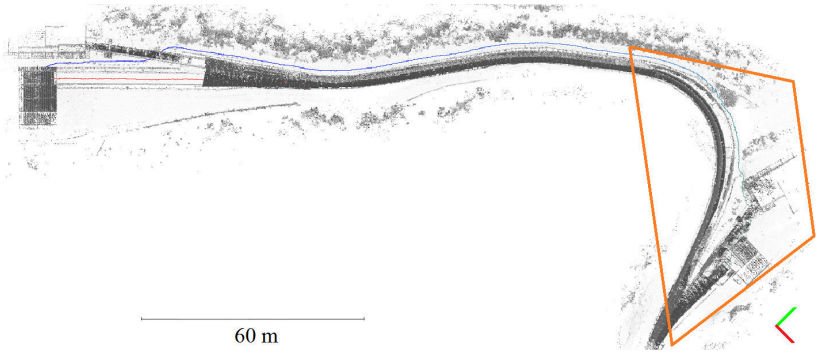


(b) A close-up of the first 350 m. The distances from the $Lil350B_{sim}$ alignment to the reference are not correlated to the residuals from the similarity transformation due to the test curve generation process.



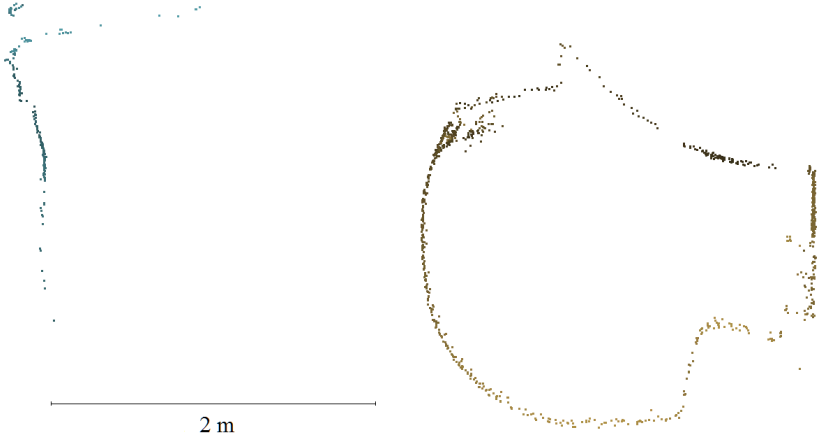
(c) Plan view.

Figure 10.12: Comparisons between the Lillehammer track alignments that were created from various sources (first 800 m).

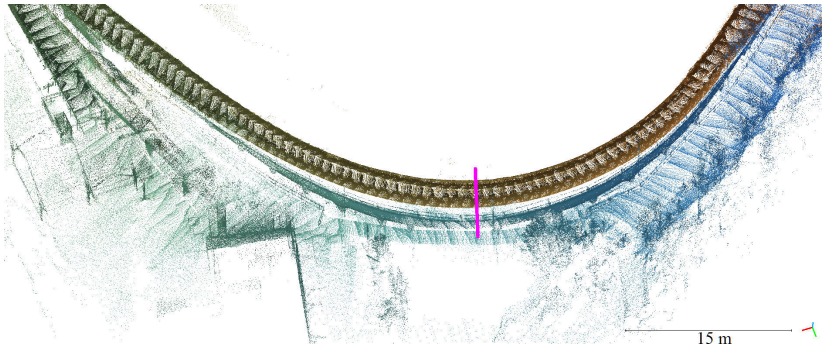


(a) Top view of the Zeb1 dataset 1; the sensor trajectory is colour-coded according to scan time.

— The footprint of the oblique view shown in figures 10.13c and 10.13d.

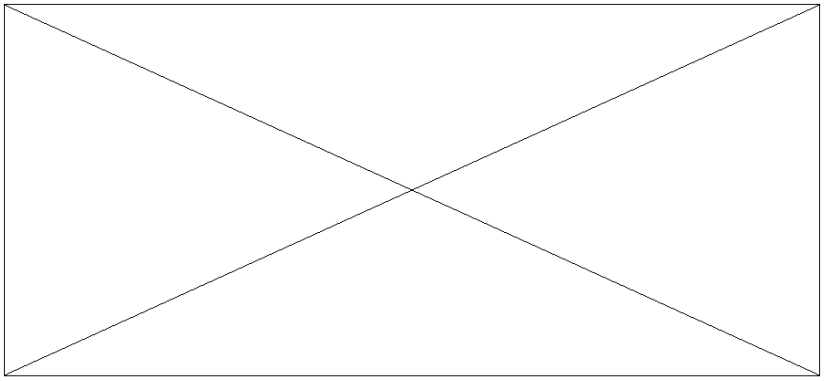


(b) Point cloud slice.



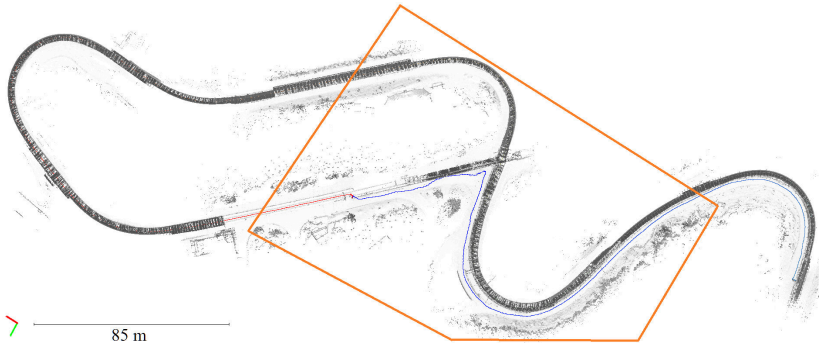
(c) Oblique north view of the colour-coded point cloud.

— The location of the slice extraction, q.v. figure 10.13b.



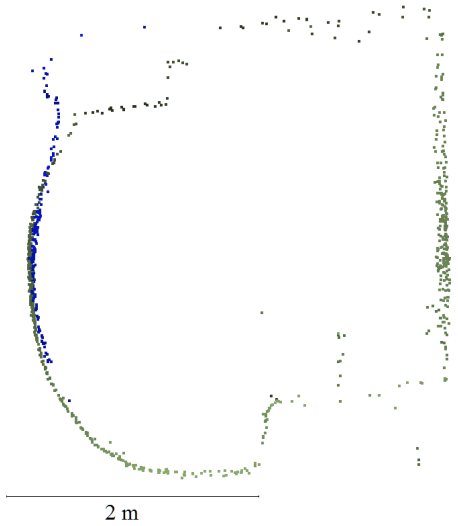
(d) Oblique north view of bend 3, q.v. table 10.4; its footprint is indicated by the orange line in figure 10.13a; reproduced from ©BLOM AS (2017) via ©Opplysningen (2017)

Figure 10.13: The internal accuracy of a 0.5 km long Zeb1 loop (*Zeb*1*s*1) at the Lillehammer track. Copyrighted material removed.

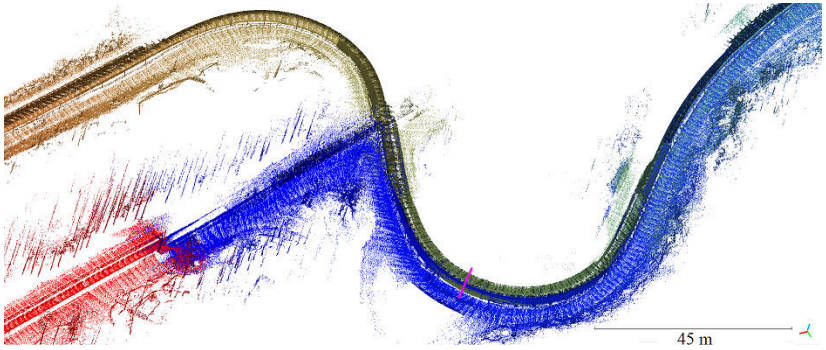


(a) Top view of the Zeb1 dataset 1; the sensor trajectory is colour-coded according to scan time, q.v. figure F.2 .

— The footprint of the oblique views shown in figures 10.14c and 10.14d.

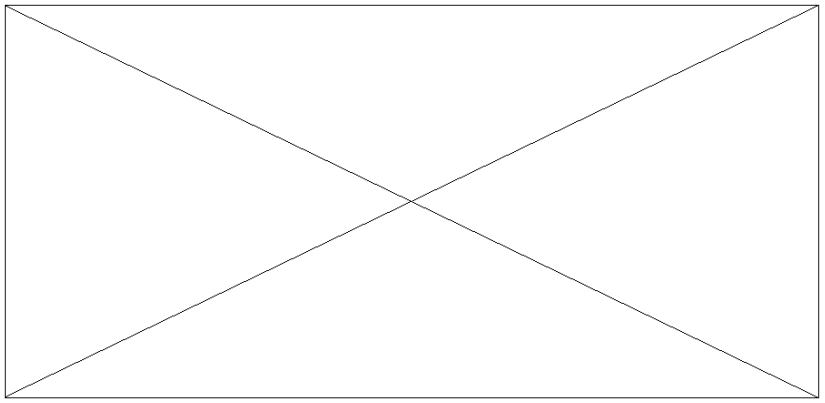


(b) Point cloud slice.



(c) Oblique north view of the colour-coded point cloud.

— The location of the slice extraction, q.v. figure 10.14b.



(d) Oblique north view of bend 12, q.v. table 10.4; its footprint is indicated by the orange line in figure 10.14a; reproduced from ©BLOM AS (2017) via ©Opplysningen (2017)

Figure 10.14: The internal accuracy of a 1 km long Zeb1 loop (*Zeb*1*s*3) at the Lillehammer track. Copyrighted material removed.

Table 10.6: Comparison of the arc radii of the first 4 bends of the Lillehammer track, as they were estimated with the proposed workflow using the 3D mesh that was generated from datasets $Lil350B_{cba}$ and $LilZeb1s4$.

		Clothoid Spline Arc Radii (m)			
		Bend ID			
DataSet ID	Fitting Parameters	1	2	3	4
$Lil350B_{cba}$	D^{cloth} = 30 m, σ_n = 0.1 m	87.14	69.73	23.16	21.46
$Lil350B_{cba}$	D^{cloth} = 25 m, σ_n = 0.1 m	89.86	71.55	24.03	21.01
$LilZeb1s4$	D^{cloth} = 25 m, σ_n = 0.1 m	76.93	62.13	22.64	23.91

Figures 10.15c and 10.16c show the clothoid splines that were fitted to the extracted track centreline using D^{cloth} = 25 m and σ_n = 0.1. For $LilZeb1s4$, bends 9, 10, and 11 could not be segmented with D^{cloth} = 30 m, due to oversmoothing; the fact that they have the same curvature sign prevented the detection of a zero crossing between them. The maximum deviation of the estimated arc radii for the first 4 bends for $Lil350B_{cba}$, between two runs using both values for D^{cloth}, was less than 4%, q.v. table 10.6. Considering their average as a reference for each radius, the maximum deviation or their estimation from the $LilZeb1s4$ was 13% for bend 1. The 95th percentile of the point-to-point distances between the two generated track models that were estimated on a 5×5 cm grid after a best fit registration using the ICP algorithm (Besl et al., 1992), was 1 m, q.v. figure 10.17.

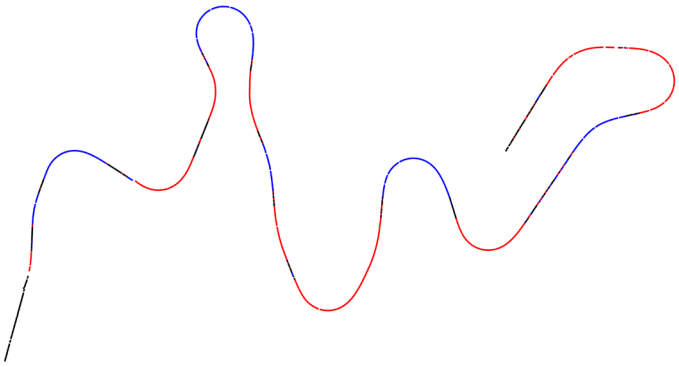
Chapter Summary

The Lillehammer track scanning case study was partially successful as contiguous segments up to 500 m long were reconstructed based solely on high resolution images. Nevertheless, a single 3D model of the entire length of the track could not be retrieved from the image-based process, as the image orientation process failed at more than one positions along the track. The failure was primarily attributed to the extremely shiny ice at several parts of the track combined with longer than 4 m along-track image acquisition intervals—a stereo image pair was acquired at each step. Due to poor satellite signal reception, the distribution of the measured GCPs did not allow for the direct georeferencing of all the track segments. In addition, there was no indication that an online feedback mechanism, based on image quality metrics, could be more effective than the judgement of the camera operator towards preventing failure by reducing the acquisition intervals.

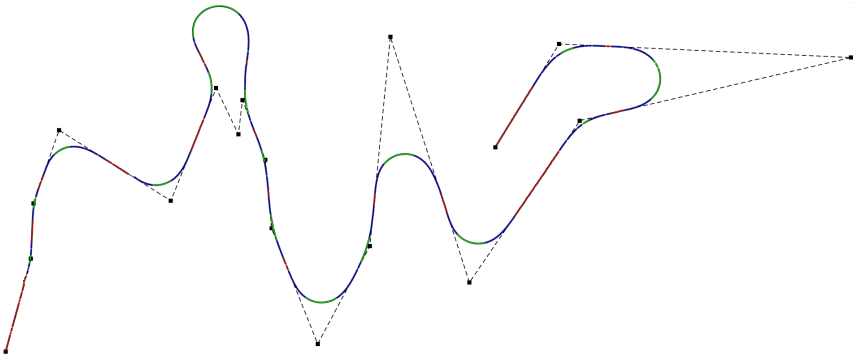
The entire length of the track was reconstructed from a video sequence and the Zeb1 laser



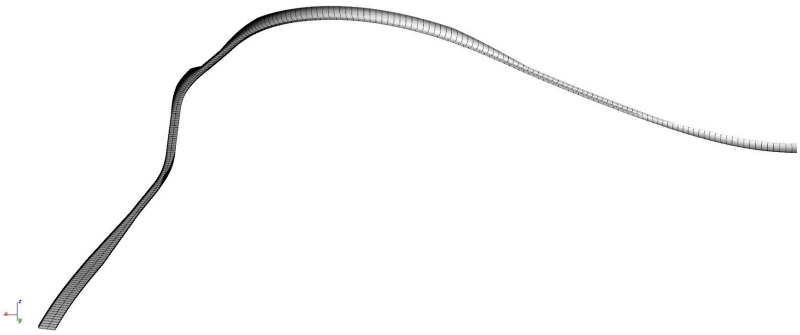
(a) The mesh was generated in Geomagic Studio (2012).



(b) The detected template points that correspond to the track's centreline. The points along the track where a full, left, right template was detected are marked in black, red, and blue, respectively.

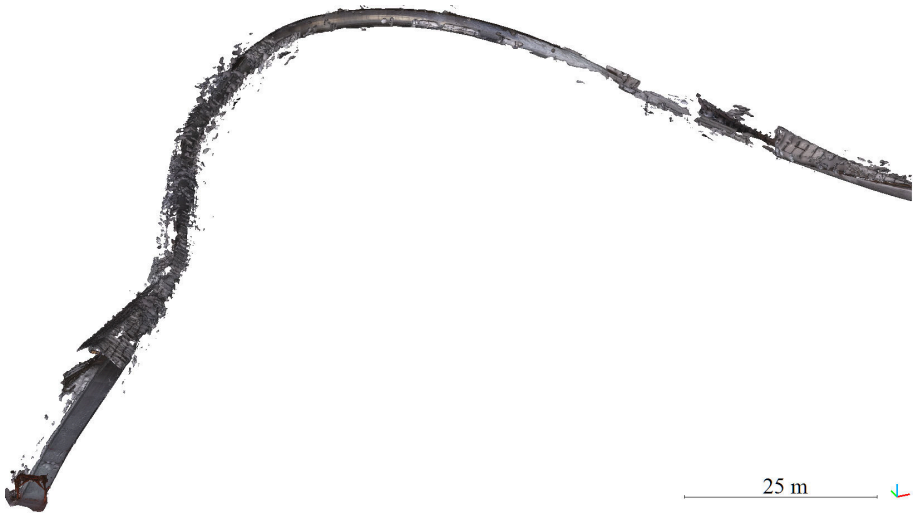


(c) The clothoid spline that was fitted to the extracted points.

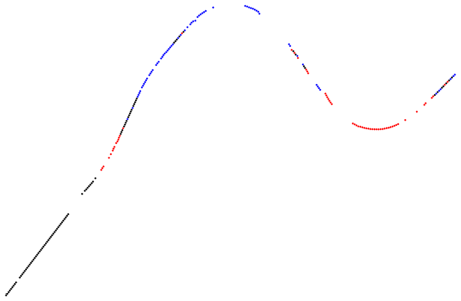


(d) The loft was generated in ©Rhino 5 (2018) from the cross sections that were extracted from the sweeping-and-morphing algorithm.

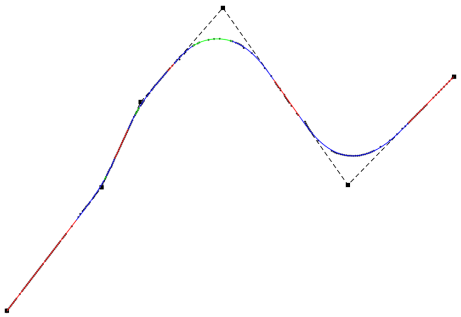
Figure 10.15: Automatic track alignment and surface extraction from dataset *LilZeb*1*s*4 (D^{cloth} = 25 m, σ_n = 0.1 m).



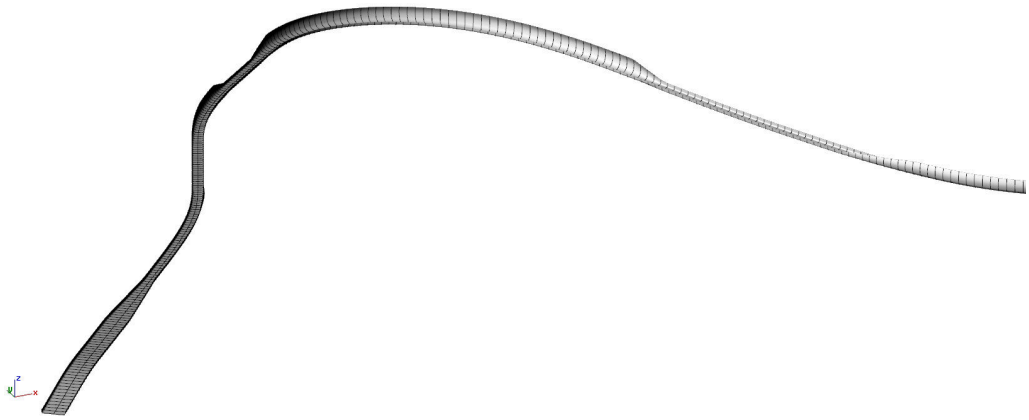
(a) The mesh was generated in PhotoScan; the surrounding environment was manually removed.



(b) The detected template points that correspond to the track's centreline. The points along the track where a full, left, right template was detected are marked in black, red, and blue, respectively.



(c) The clothoid spline that was fitted to the extracted points.



(d) The loft was generated in ©Rhino 5 (2018) from the cross sections that were extracted from the sweeping-and-morphing algorithm.

Figure 10.16: Automatic track alignment and surface extraction from dataset $Lil350B_{cba}$ (D^{cloth} = 25 m, σ_n = 0.1 m).

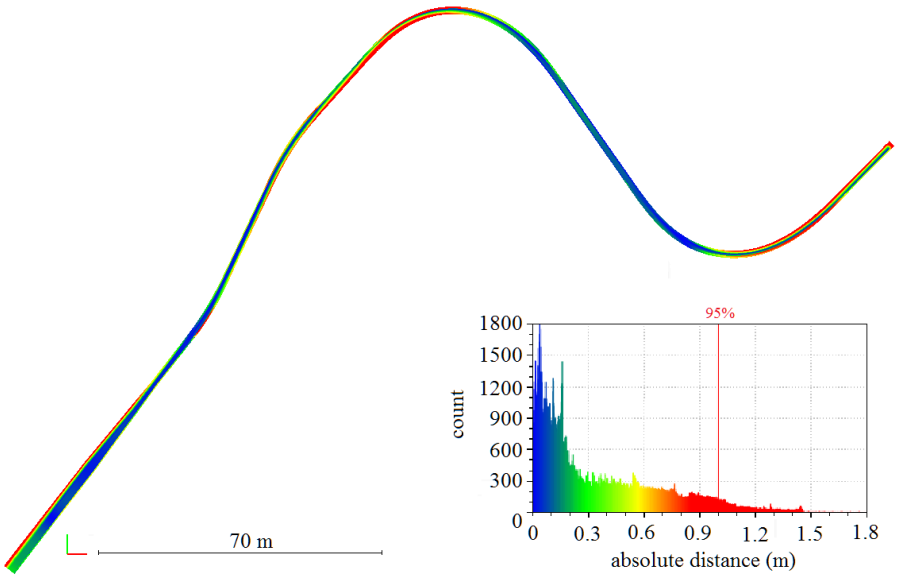


Figure 10.17: The track surface that was generated from dataset *LilZeb*1*s*4, colour-coded according to the point-to-point distances on a 5×5 cm grid from the one that was generated from $Lil350B_{cba}$; The distances for 95% of the points where less than 1 m.

scanner. Due to the low resolution of the former, the accuracy of the result was very poor, especially when no GCPs were used during the image alignment process. The latter returned a complete point cloud that was locally dense and reliable. However, the along track positional drift was significant and could only be retained within similar limits to the photogrammetric solution only by splitting the track to several overlapping segments, in order to keep the scan time of each one below 20 min. One downside of this approach is that the total scan time may increase, potentially beyond what would be required for image acquisition.

Chapter 11

Igls Track Scanning

A second case study was conducted in Igls, q.v. figure 11.1 and table 2.1, in order to test three hypotheses. Firstly, that given 1 hour on the site, the entire length of a 1300–1400 m long ice track can be reconstructed by a high resolution hand-held camera if the image acquisition intervals are kept strictly below 4 m. Secondly, that without a survey control network, the shape of the track can be reliably reconstructed and that its datum can be defined based on knowledge about the geometry of the track's structure itself. Lastly, that the proposed geometric processing algorithm is able to extract and approximate the full extents of the track's surface, without user-intervention, starting from an incomplete, noisy, and unsegmented 3D mesh, which was generated from dense image matching.

Section 11.2 describes the experiments on the ice track, whereas section 11.1 is concerned with image-based 3D scanning and modelling of the concrete structure of the track without the ice layer. The results from the automatic track surface generation are presented in section 11.3. In order to reference the 3D models to a local reference system and assess the geometric accuracy of the results, 18 ground control points were measured on site, following the workflow that was described in section 10.1.1, q.v. figure 11.3 and table E.2.

11.1 Concrete Track Scanning

In order to generate a reference 3D model for the track, its concrete structure was reconstructed from image sets that were collected during an off-season scan:

- A Canon EOS5D Mark II DSLR camera with a Canon EF 28 mm prime lens was used to collect 936 images with a resolution of 21 MP, q.v. table D.1, in JPG format according to the imaging configuration of generic block 5 at the standing position, within 105 min. This

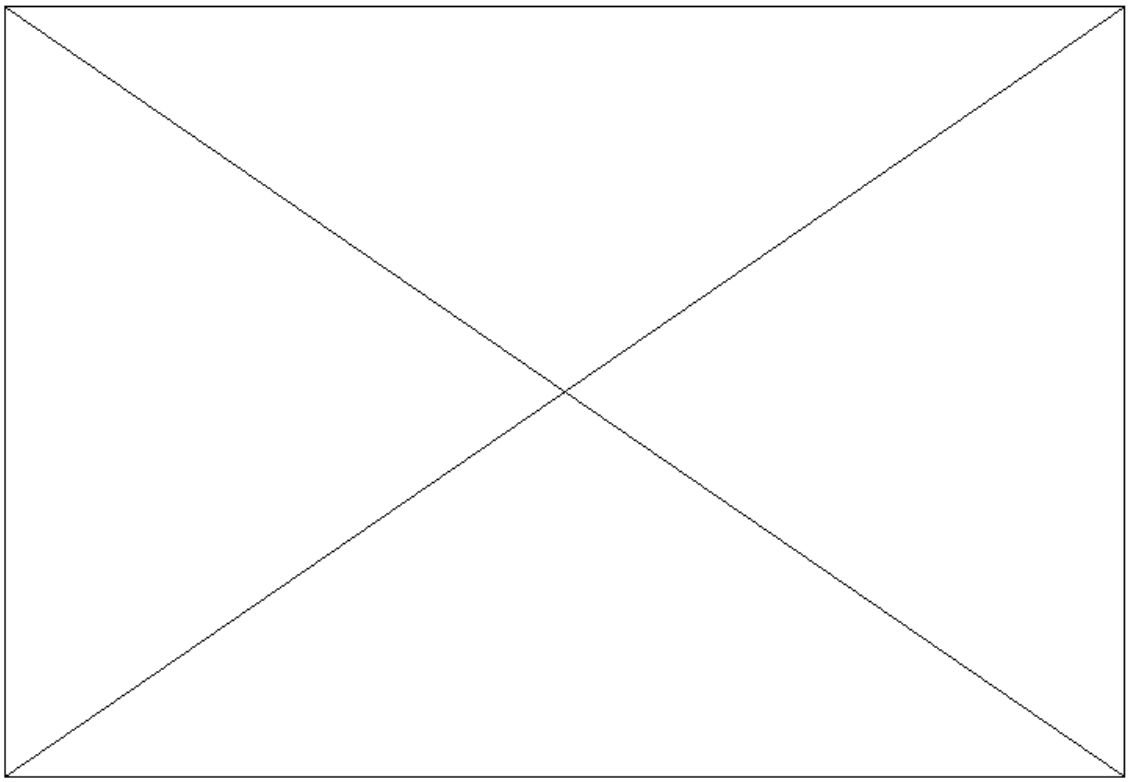


Figure 11.1: A north-east oblique view of the Igls track; reproduced from ©BLOM AS (2017) via Bing Maps (2017). Copyrighted material removed.



Figure 11.2: Views of the Igls concrete track on the day it was scanned.

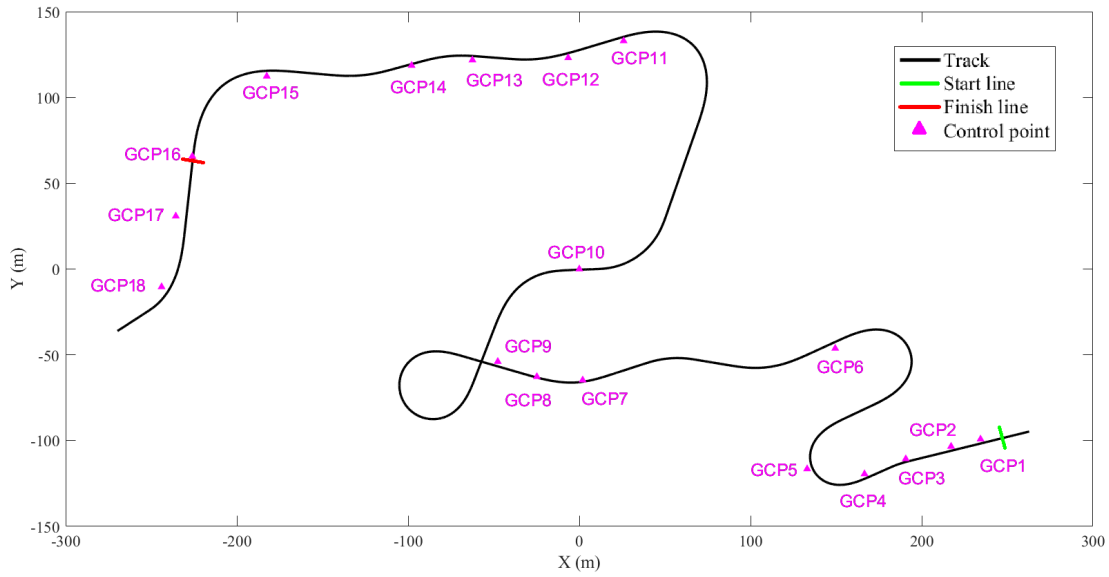


Figure 11.3: Plan view of the the Igls track with the ground control point locations.

image set will be referred to as *IglsConcC*5.

- A Nikon D300 DSLR camera with a 28 mm prime lens was used to collect 730 images with a resolution of 12 MP in JPG format, q.v. table D.3, according to the imaging configuration of generic block 11 at the standing position, within 77 min. This image set will be referred to as *IglsConcN*11.

The across-track intervals between successive camera stations for *IglsConcC*5 and *IglsConcN*11 were on average 1.45 m and 3.75 m, respectively. The across track half-base for *IglsConcC*5 and the stereo base of the image pairs for *IglsConcN*11 were 0.3 m and 0.85 m, respectively, q.v. figure G.1 and G.2.

There were no significant problems during the image orientation process due to the good lighting conditions and reflectance properties of the track's surface; the fact that the blinds were up, allowed more feature points to be tracked along longer image sequences. The total RMSE of the similarity transformation of the SfM solution using all the available GCPs for the *IglsConcC*5 and *IglsConcN*11 image sets were 0.35 m and 0.29 m, respectively. The total RMSE of a constrained self-calibrating bundle adjustment for both *IglsConcC*5 and the combined image set (*IglsConcCN*), q.v. figure 11.4, using 9 GCPs at approximately 150 m intervals along the track—namely 1, 5, 6, 8, 10, 11, 13, 15, and 18—was below 0.05 m at the control and check points, q.v. figure G.3.

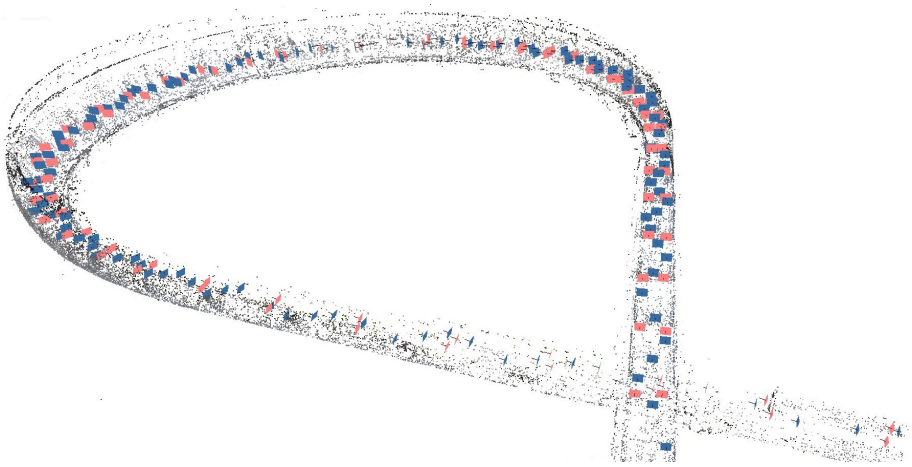


Figure 11.4: The implementation of generic block 5 (blue) and 11 (red) at the Igls track.



Figure 11.5: Views of the Igls ice track on the day it was scanned.

11.2 Ice Track Scanning

The track in the iced state, q.v. figure 11.5, was scanned on a mostly cloudy November day at noon with the temperature at $10°C$, using a Canon EOS5D Mark II DSLR camera with a Canon EF 28 mm prime lens, q.v. table D.2. The concrete structure of the track was covered with a frosty layer of ice. Approximately 20% of the track's length was roofed and for almost 20% of it the blinds were limiting the view to the surrounding environment, q.v. figure 11.7. Along an approximately 1350 m long stretch, comprising 15 bends, 555 still frames with 21 MP (5616×3744) resolution were collected within 60 minutes—the average scanning pace was 6.5 sec/image or 22.5 m/min—according to the geometry of block 26, q.v. figure 11.6, and the guidelines that were described in section 9.2; this image set will be referred to as $IglsIce26$. The camera settings, across-track half-base, and along-track distances for each camera station with respect to its position along the track are shown in figure G.4; the latter were 2.75 m, on average, and not greater than 4 m.

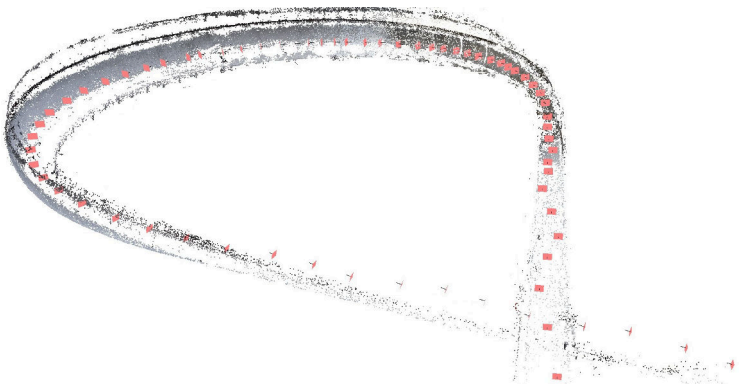


Figure 11.6: The implementation of generic block 26 at the Igls track.



(a) The entrance to bend 7 (the kreisel).



(b) The entrance to bend 8.

Figure 11.7: Transitions from open to roofed segments along the Igls track pose a challenge to the image orientation process as the overlapping area is reduced and the light variations may lead to suboptimal image exposure.

11.2.1 Image Orientation

The relative orientation of the images at their full 21 MP resolution in a single automatic step was problematic for this dataset; the problems were similar to those discussed in sections 5.2 and 10.1.3. In this case, the problems occurred at the transitions from the uncovered to covered parts of the track because not only did the blinds prevent the detection of potentially corresponding points between the images but also created a challenging situation with respect to the required image exposure balance between very bright and very dark areas on the surroundings and the surface of the track, respectively, q.v. figure 11.7. In addition, even for the attempts that generated enough feature matches, if the SfM process was applied on the entire image set, a sparse reconstruction could only be partially retrieved.

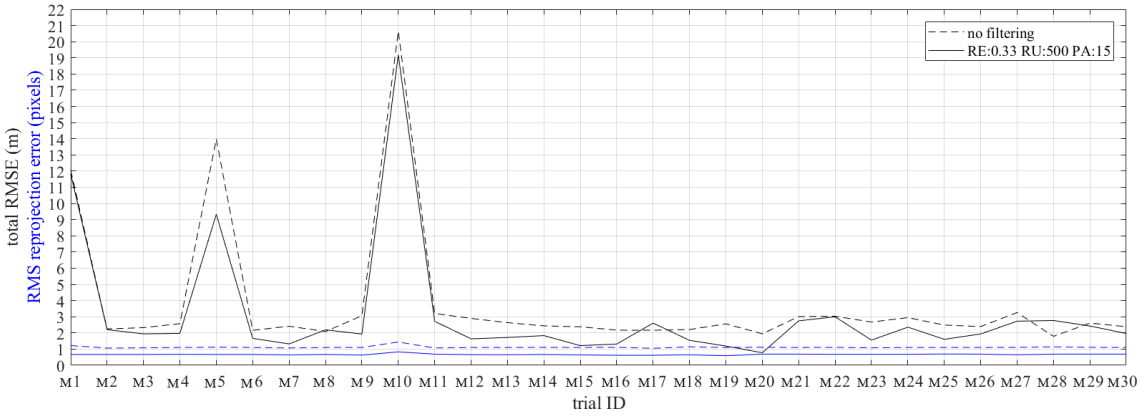


Figure 11.8: The RMS reprojection errors and total RMSE at the control points after a similarity transformation of the SfM solution of 30 image orientation trials of image set *IglsIce*26 at 1/4 of its original resolution.

In order to circumvent these problems, and identify potential solutions, the image set was processed at 3 resolution levels using two strategies. Initially, 30 trials were conducted at 1/4 of the original image resolution. The image matching and orientation process lasted approximately 30 minutes for each trial, with an average number of 30000 features detected on each image. Half of the trials were successful in automatically estimating the relative orientation of the entire set within a single network; the other half were also successful but required manual processing. The total RMSE at the control points of a similarity transformation of the SfM solution using all GCPs, was less than 3.3 m for 27 out of the 30 trials. The average RMSE at the control points for the best 27 trials was 2.85 m, if all the feature matches were considered during the SfM solution, and was reduced to 2.30 m after filtering potential outliers, q.v. figure 11.8. The total RMS reprojection errors for the bundle adjustments with and without manually removing outliers were, on average, 1.11 and 0.67 pixels, respectively.

Table 11.1 summarises the results of 10 additional trials which successfully recovered the relative orientation of the entire image set at their full resolution or upsampled by a factor of 4; the average duration of the orientation process was 4 hours for each one. For the trials that are described as nonsequential—namely A, B, and C—the relative orientation of the entire image set was retrieved in a single step without user intervention. For the other 7 trials, the relative orientation of image subsets, comprising approximately 100 images each, were processed sequentially from start to finish. The relative orientation of the problematic image sequences that are depicted in figure 11.7, was estimated at a lower resolution than the rest. More specifically, the relative orientation of the sequence that is depicted in figure 11.7a could not be estimated for the upsampled images but

was successful at full resolution. Similarly, the relative orientation of the sequence that is depicted in figure 11.7b could only be estimated if the original images were subsampled by a factor of 4.

The SfM solution for each network was calculated with 5 different filtering threshold combinations. The result was not necessarily improved by eliminating potential outliers, as the increased number of detected features at these resolution levels returned better RMS reprojection errors; setting the upper bounds for reprojection error, reconstruction uncertainty, and projection accuracy to 0.33 pixels, 500, and 15, respectively, appeared to provide a good balance—this particular solution for trial C will be referred to as $IglsIce26C_{sim}$. The improvement to the total RMSE at the control points of a similarity transformation of the SfM solution using all GCPs, compared to that of the 30 trials that were conducted with the subsampled image set, q.v. figure 11.8, was significant—almost 10 times better, with 4 networks having RMSE below 0.3 m—yet inconsistent, as the results of trial B were in the 2.48–8.25 m range. Figure 11.9 illustrates the control point coordinate residuals of self-calibrating bundle adjustmens constrained by GCPs at increasing intervals along the track—starting from the same SfM solution. The almost linear relation between the maximum residual $max\Delta$ and the average along-track GCP interval $meanGCPint$ can be summarised—even though it should not be generalised—as $max\Delta = 0.05 + 0.05 meanGCPint/100$ m.

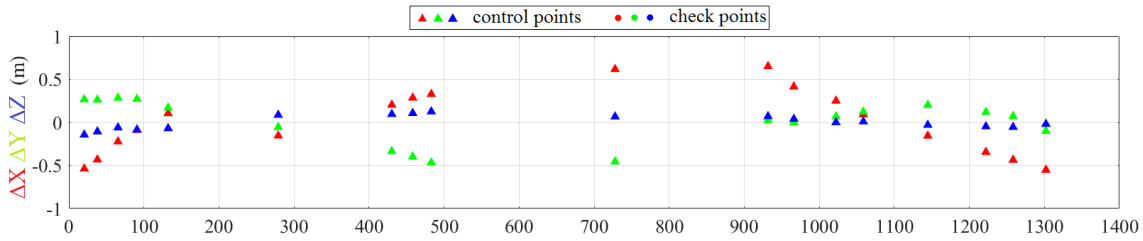
11.2.2 Image Segmentation

In an attempt to segment the track on the images in order to facilitate the dense reconstruction process, image masks were automatically generated for each image, using the process that was described in section 7.2 using the SfM solution for $IglsIce26C_{sim}$. In order to assess the effectiveness of the segmentation process, the masks for 290 of the images were created manually, q.v. figure 11.11a. Even though the recall (total number of correctly detected foreground pixels/true foreground pixels) was greater than 94% for 94% of the images, the average precision (total number of correctly detected foreground pixels/total detected foreground pixels) and accuracy (total number of correctly detected foreground and background pixels/correctly and incorrectly detected foreground and background pixels) were 71% and 82%, respectively. Only 5% of the masks classified more than 10% of the track in the background, which means that a portion of these parts can potentially be reconstructed from other overlapping frames. In contrast, even though the area of the frame that was identified as foreground covered 60% of the frame, on average, it was 1.4 times greater than the actual area of the track's projection on the images, q.v. figure 11.11c. Therefore, even though a large part of the background was not included in the point cloud, the track's edges were not be sharply segmented. Nevertheless, both the start line endpoints, which are sought for in

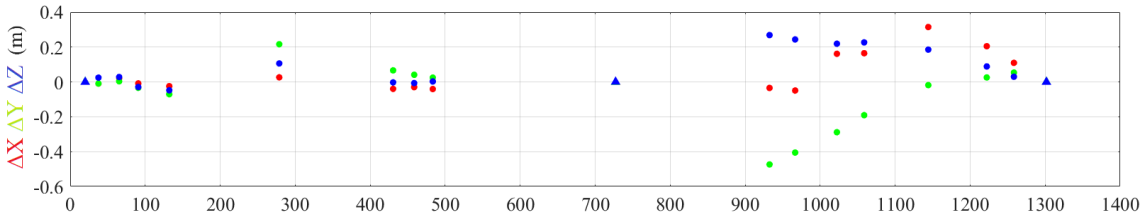
Table 11.1: Image orientation report for 10 trials in PhotoScan which successfully recovered the relative orientation of the entire image set of the Igls ice track. The images were processed at their full resolution or upsampled by a factor of 4, in a single set or split in sequential subsets in order to estimate their relative orientation incrementally from start to finish; 5 filtering schemes were used for outlier removal.

Net ID	Sequential	Resolution	Detected Feature Limit	Matched Feature Limit	Max Threshold Values[a] RE	RU	PA	Total Number of Points	RMS RE[a] (pixels)	GCP RMSE (m)
					0.81	1000	110	552685	0.67	0.92
					0.50	700	30	547645	0.64	0.93
A	no	full	-	4000	0.33	500	15	498368	0.51	0.19
					0.20	300	15	393851	0.38	2.26
					0.10	150	10	208442	0.20	3.88
					0.46	992	110	540973	0.64	8.25
					0.50	700	30	536695	0.61	2.99
B	no	full	-	4000	0.33	500	15	505569	0.58	2.48
					0.20	300	15	392497	0.38	5.68
					0.10	150	10	207421	0.20	6.96
					0.78	1000	111	911656	0.62	0.45
					0.50	700	30	896567	0.58	0.32
C	no	full	-	-	0.33	500	15	833520	0.50	0.45
					0.20	300	15	649970	0.38	0.23
					0.10	150	10	323237	0.19	0.52
					0.64	1000	110	549455	0.64	0.65
					0.50	700	30	545141	0.61	0.67
D	yes	full	-	4000	0.33	500	15	506873	0.51	0.85
					0.20	300	15	392313	0.37	1.75
					0.10	150	10	210155	0.21	0.66
					0.75	1000	166	859136	0.60	0.41
					0.50	700	30	844820	0.57	0.45
E	yes	full	-	-	0.33	500	15	789428	0.49	0.64
					0.20	300	15	619256	0.38	0.29
					0.10	150	10	321149	0.20	0.62
					0.80	1000	143	883495	0.59	0.60
					0.50	700	30	869058	0.56	0.69
F	yes	full	-	-	0.33	500	15	810568	0.49	0.66
					0.20	300	15	637644	0.37	0.75
					0.10	150	10	331706	0.20	0.61
					1.70	1000	144	939819	0.63	0.33
					0.50	700	30	928057	0.59	0.38
G	yes	full	-	-	0.33	500	15	858093	0.50	0.49
					0.20	300	15	668835	0.37	0.49
					0.10	150	10	331267	0.18	0.43
					0.44	1000	111	727303	0.55	0.17
					0.50	700	30	719345	0.52	0.17
H	yes	x4	100000	-	0.33	500	15	686809	0.46	0.30
					0.20	300	15	535796	0.34	0.44
					0.10	150	10	276663	0.18	0.32
					0.61	1000	144	699921	0.46	1.37
					0.50	700	30	691235	0.44	1.40
I	yes	x4	150000	-	0.33	500	15	658950	0.40	1.36
					0.20	300	15	524355	0.31	2.30
					0.10	150	10	277697	0.17	1.40
					0.52	1000	144	947375	0.45	1.14
					0.50	700	30	935791	0.44	1.18
J	yes	x4	250000	-	0.33	500	15	875349	0.39	0.89
					0.20	300	15	679448	0.30	1.01
					0.10	150	10	329759	0.16	1.22

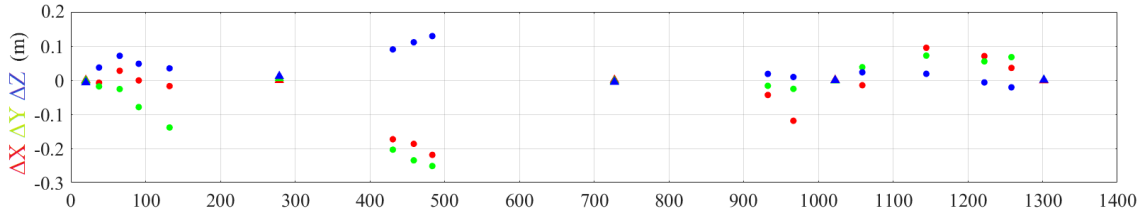
[a] RE: Reprojection Error (pixels) RU: Reconstruction Uncertainty PA: Projection Accuracy



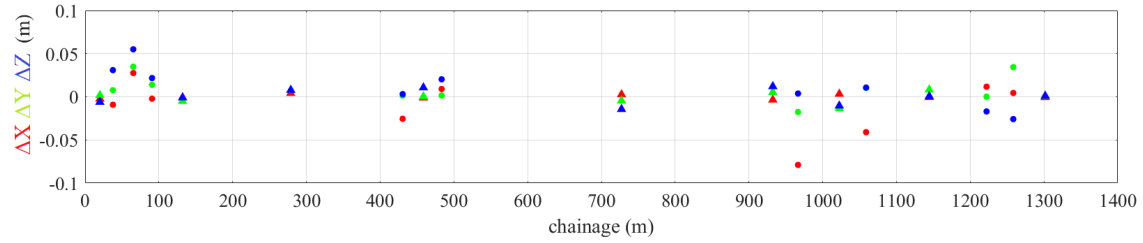
(a) *IglsIce*26*C*$_{sim}$: Residuals of the similarity transformation using all the available GCPs.



(b) Residuals of a fully constrained bundle adjustment of *IglsIce*26*C*$_{sim}$ using GCPs 1, 10, and 18 as control points.

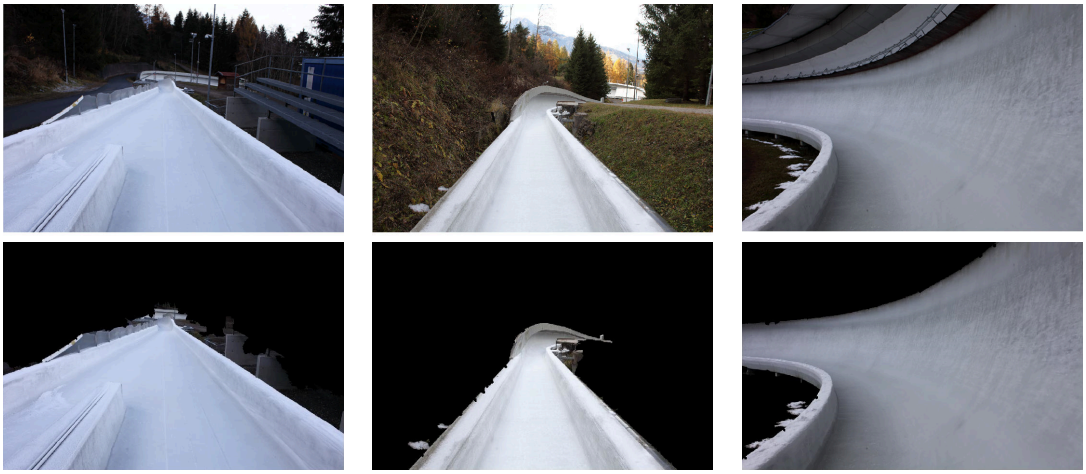


(c) Residuals of a fully constrained bundle adjustment of *IglsIce*26*C*$_{sim}$ using GCPs 1, 6, 10, 13, and 18 as control points.



(d) Residuals of a fully constrained bundle adjustment of *IglsIce*26*C*$_{sim}$ using GCPs 1, 5, 6, 8, 10, 11, 13, 15, and 18 as control points.

Figure 11.9: The control and check point coordinate residuals for the similarity transformation and constrained bundle adjustment of dataset *IglsIce*26*C*.



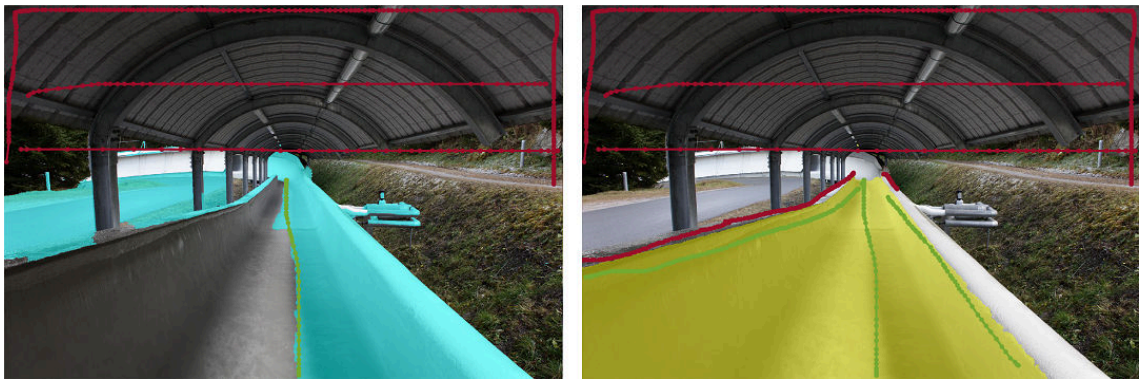
(a) Starting straight. **(b)** A straight segment. **(c)** The kreisel bend.

Figure 11.10: Views of the Igls track before (top) and after (bottom) the image segmentation of dataset $IglsIce26$.

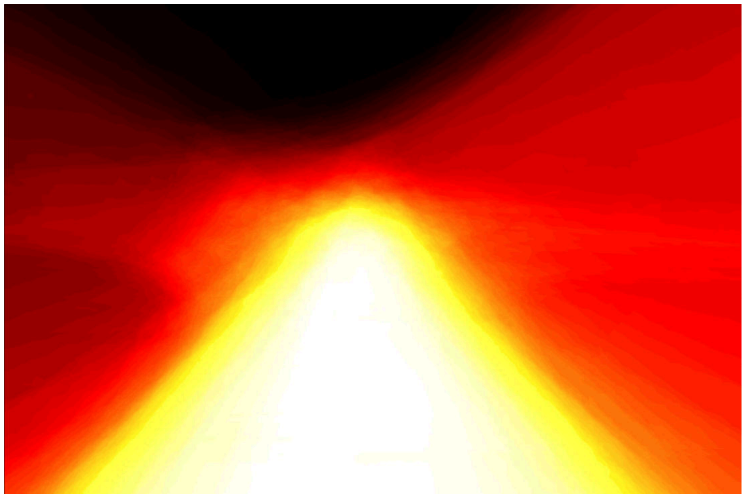
the foreground part of the first few frames, were successfully detected; the along track positional error of the start line detection process was 15 cm, q.v. figure 11.12. This was considered to be an acceptable approximation which should be subsequently refined within a more narrowly focused search by back-projecting the endpoints on the images.

11.2.3 Dense Reconstruction

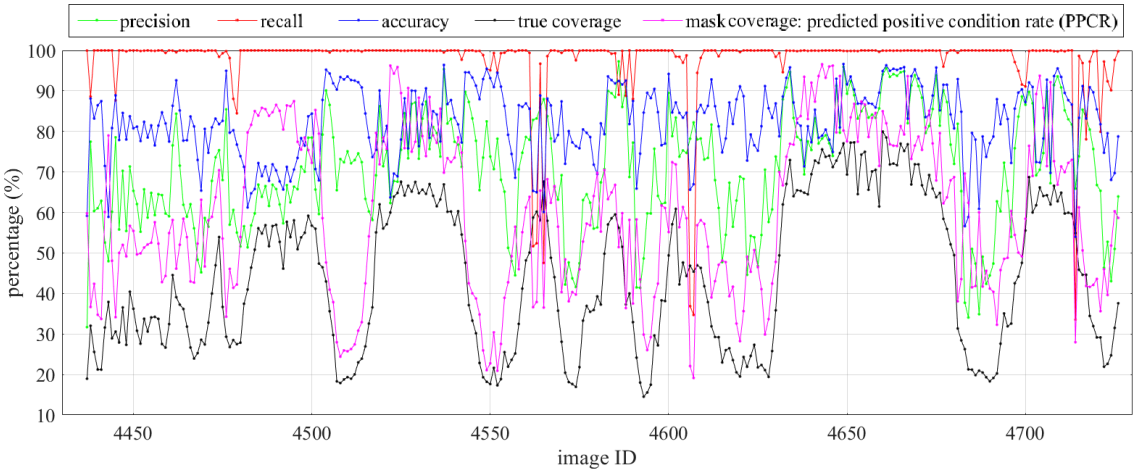
Dataset $IglsIce26C_{sim}$ was processed further in order to assess different strategies for dense image matching and 3D mesh reconstruction within PhotoScan, with respect to their efficiency and effectiveness in generating the full extents of the surfaces that are depicted on overlapping parts of the images of the track site and reduce potential gaps without introducing undesired artifacts. As discussed in sections 3.3.3 and 3.6.1, the dimensions of the scene's bounding box can implicitly affect both the point cloud and mesh generation. In general, the bounding volume definition can be based on the sparse point cloud, which comprises the 3D locations of the feature matches on the object of interest. In this case, the bounding box can be defined with respect to the image locations, as well, q.v. figure 11.13. However, there are several implications with respect to the use of a forward-facing camera inside a track for the generation of a virtual environment. On the one hand, the bounding box dimensions of a typical track site, which can be close to $500 \times 500 \times 50$ m, are not representative of its shape and size. On the other hand, any object within this box may be visible on any image, which means that the depth range for image matching may be significantly overestimated and delay the dense matching process.



(a) The reference masks were created manually in MATLAB (2017) using the interactive "Lazy Snapping" algorithm. The selection of the appropriate foreground (green) and background (red) strokes was not always straightforward.



(b) A heat map was created by aggregating the manually generated masks. The black area marks the part of the frame where the track was not depicted on any of the images.The slight imbalance between the left and right edges is attributed to the kreisel being a left bend.



(c) Precision, recall, and accuracy of the image segmentation in foreground (track) and background.

Figure 11.11: Assessment of the automatic track segmentation process in image space.



Figure 11.12: Igls track start line detection: the true (green) and the detected (red) start line at the Igls track; the along track detection error was 15 cm.

The bounding volume could be split in smaller regions, so that only those close to one of the images—which would also include the track—could be processed independently. However, not only the track itself is of interest but also what is visible from the athlete's point of view during a sled run. Even though most of the track's bounding box volume is not visible on the images, especially for fully covered tracks, and may not be visible from the athletes point of view during a simulated run either, the decision on which parts are important and should be included in order to be used for the generation of an immersive virtual environment, is not straightforward. Suppressing geometric information that lies farther away from the track, regardless of the visual cue it provides, should only be considered if it results to a significant reduction in processing time and effort.

Either way, such bounding volumes are not expected to segment the track's surface from its surrounding environment so that it can directly feed the physics engine of a simulating system or be independently processed further to this end. Thus, the use of image masks was also considered, q.v. sections 3.5.1 and 11.2.2. In the first instance, a qualitative assessment of the generated geometry with respect to the reconstructed detail, noise level, and completeness was conducted based on visual inspection. The processing times are also reported for reference—all experiments were processed on the same laptop with all its resources dedicated to the reconstruction task; for its technical specifications see section 3.6.1.1.

11.2.3.1 Processing the Bounding Box of the Track Site in a Single Step

The first set of dense reconstructions was based on a single bounding box which included the entire site, q.v. figure 11.13a. Dense image matching was conducted using the images at 1/16, 1/64,



(a) A single bounding box including the entire site.



(b) Partitioning of the track site into 66 parts along the track length length.

Figure 11.13: A sparse point cloud of the Igls track and two options for the definition of the bounding box that was used during the 3D point cloud and mesh generation.

and 1/256 of their original resolution. Two levels for the depth map filtering, namely moderate and aggressive, were considered for the two lower resolutions. The higher resolution level was only processed with the aggressive filter, as the amount of time which was required, i.e. more than a day, for the generation of a point cloud with a practically unmanageable size from the noisy depth maps was prohibitive. A 3D mesh, comprising 1 triangle face for every 5 cloud points, was generated from each point cloud. The duration of each dense image matching, point cloud generation, and mesh reconstruction process is summarised in table 11.2; for the latter, it depended on the size of the input dense point cloud, and not the number of generated faces, as the entire scene is reconstructed using all the available points and subsequently decimated to the user-defined number of mesh 3D faces. Figure G.5 shows the generated points clouds and meshes in the arbitrary photogrammetric model from the point of view of the image in figure 11.1. Figures G.6–G.11a provide closer views of the dense point clouds, which were generated with and without using image masks, and the corresponding meshes for three track segments; namely the starting straight, a straight segment, and a bend segment, q.v. figure 11.10.

Table 11.2: Dense point cloud and mesh sizes and processing times for Igls track dataset $IglsIce26C_{sim}$

		Without masks	
Resolution	Filtering	Point cloud	Mesh
1/256	Moderate	P:3642686 Dp:20/32 min	F:728537 V:380422 Dm:5 min
	Aggressive	P:3950421 Dp:18/17 min	F:790084 V:416926 Dm:6 min
1/64	Moderate	P:14781295 Dp:71/119 min	F:2956259 V:1533184 Dm:21 min
	Aggressive	P:16962513 Dp:66/43 min	F:3392501 V:1772363 Dm:25 min
1/16	Aggressive	P:71345454 Dp:248/267 min	F:14269089 V:7230709 Dm:63 min
		With masks	
Resolution	Filtering	Point cloud	Mesh
1/256	Moderate	P:4112460 Dp:29/32 min	F:822491 V:436573 Dm:7 min
	Aggressive	P:4058769 Dp:28/24 min	F:811752 V:430538 Dm:6 min
1/64	Moderate	P:18272469 Dp:75/162 min	F:3654493 V:1866508 Dm:14 min
	Aggressive	P:19845877 Dp:73/60 min	F:3969174 V:2011972 Dm:14 min
1/16	Aggressive	P:84264920 Dp:247/294 min	F:16852983 V:8461570 Dm:21 min

P:total number of points in the dense point cloud

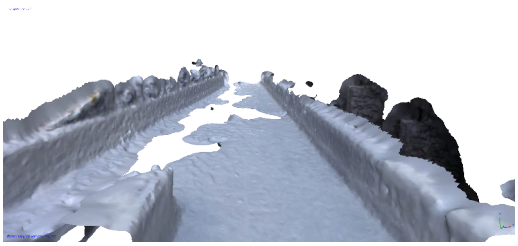
Dp:duration of the depth map/point cloud generation, Dm:mesh generation processing time

F:total number of mesh triangle faces, V:total number of mesh vertices

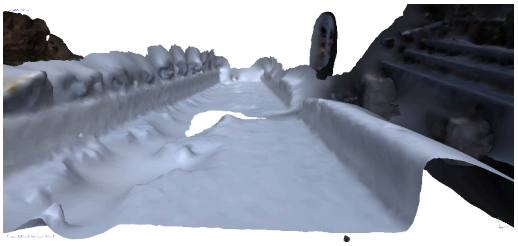
The mesh quality appears to be good at 1/16 of the original 21 MP resolution, acceptable at 1/64 and unusable at 1/256; the effect is less prominent along the curves. Small gaps in each mesh were compensated for by the default built-in interpolating algorithm; however, this process had the undesirable effect of creating additional triangles along the edges of the dense point clouds—or to be more exact, did not manage to precisely trim the generated mesh at the boundaries of the point cloud—which rendered the use of masks irrelevant. The effect of using smaller bounding boxes, in order to implicitly affect the mesh trimming process, is discussed in section 11.2.3.2. Choosing to disable the interpolating option was almost 40 times slower and returned a non-manifold mesh with a large number of disconnected triangles, which essentially added little value to the dense point cloud itself. In addition to the fact that the effect of mesh interpolation, in practice, reverted any of the track edges that were revealed from the image segmentation to their previous state, using the image masks for the dense matching did not reduce the size of the point cloud or the time required for its generation. In fact, both were increased, even without considering the time it took to generate and import the masks themselves; which was approximately 40 minutes. There was an improvement in the processing time for mesh generation, which was more evident for the highest resolution level; approximately 3 times faster.

11.2.3.2 Independent Processing of the Track Site Segments

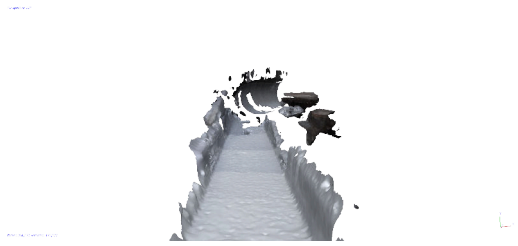
A second set of experiments was conducted on dataset $IglsIce26C_{sim}$, after manually splitting the bounding box to 66, approximately $30 \times 30 \times 30$ m, non-overlapping boxes, in order to independently reconstruct the track segment inside each box, q.v. figure 11.13b. A dense point cloud was created for each segment with the images at a quarter of their resolution and aggressive filtering. The total time for depth map/point cloud/mesh generation was reduced from 247/294/21 min and 248/267/63 min, q.v. table 11.2, to 183/47/139 min and 195/44/142 min, with and without using masks, respectively. Larger parts of the track's surface were not reconstructed, q.v. figure 11.14f, presumably as a result of the maximum gap size—or triangle edge length—being defined relatively to the size of the bounding box and not the total area of the mesh. On the other hand, using image masks was more effective—q.v. figure 11.14e, cf. figure G.11. Nevertheless, as few as 2 images where a track's surface edge is not accurately segmented are enough to reconstruct objects that are adjacent to it, q.v. figure 11.14, which means that post processing of the mesh is necessary. In addition, after merging all the independently generated meshes into one that represented the entire site, intersecting triangles at the seams need to be removed.



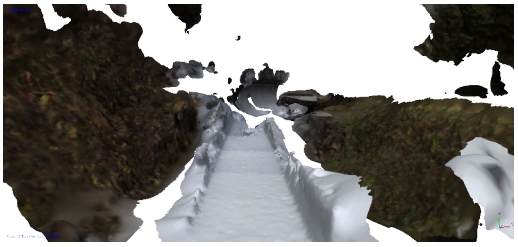
(a) The starting straight with masks.



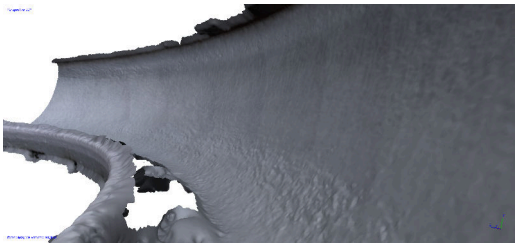
(b) The starting straight without masks.



(c) A straight segment with masks.



(d) A straight segment without masks.



(e) The kreisel bend with masks.

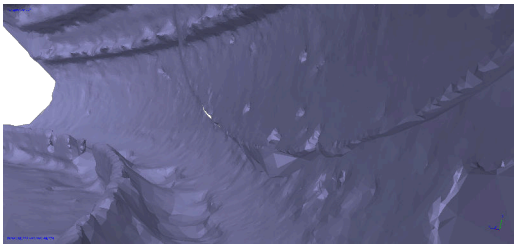


(f) The kreisel bend without masks.

Figure 11.14: The segments of the Igls track that are depicted in figure 11.10 were independently reconstructed using a smaller bounding box for each one.



(a) The starting straight that is depicted in figures G.6a and 11.10a.



(b) The bend segment that is depicted in figure G.10a and .

Figure 11.15: Visibility-consistent mesh generation in PhotoScan directly from the depth maps for dataset $IglsIce26C_{sim}$.

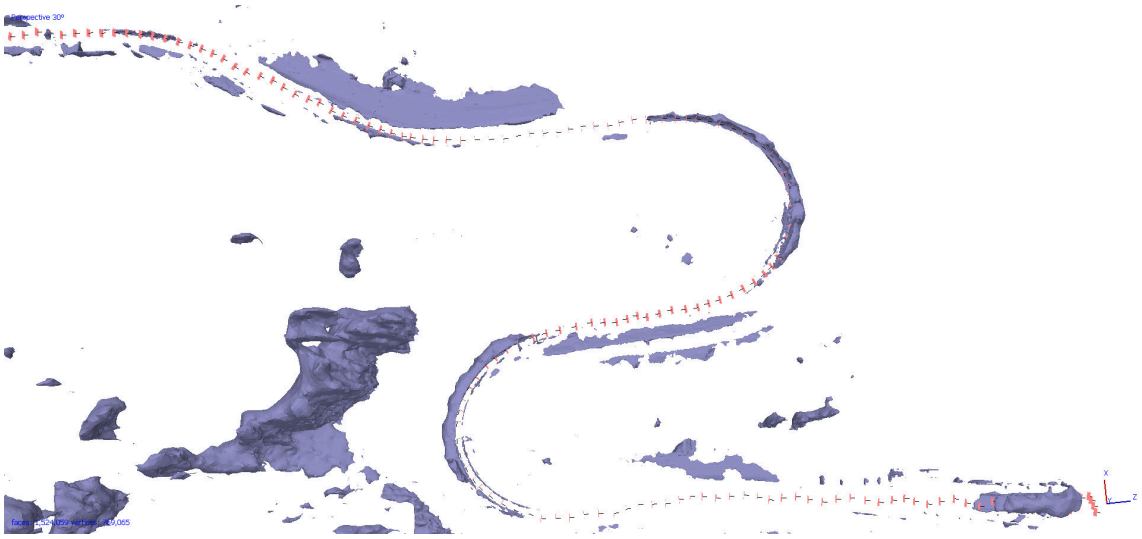
11.2.3.3 Direct Mesh Generation from the Depth Maps

Finally, the surface reconstruction method in all the aforementioned processing workflows was substituted by one that estimates the 3D mesh directly from the depth maps, without generating a point cloud. Even without using strict masks, q.v. section 3.6.1.1, only a very small percentage of the track was reconstructed, q.v. figure 11.16a, as this method is more suitable for objects that have been scanned in their entirety and their full extents are visible in every image, q.v. figure 11.16. For the parts that were successfully reconstructed using the partitioned bounding boxes, extra care was required at the seams along their boundaries as this process yields watertight surfaces, q.v. figure 11.16.

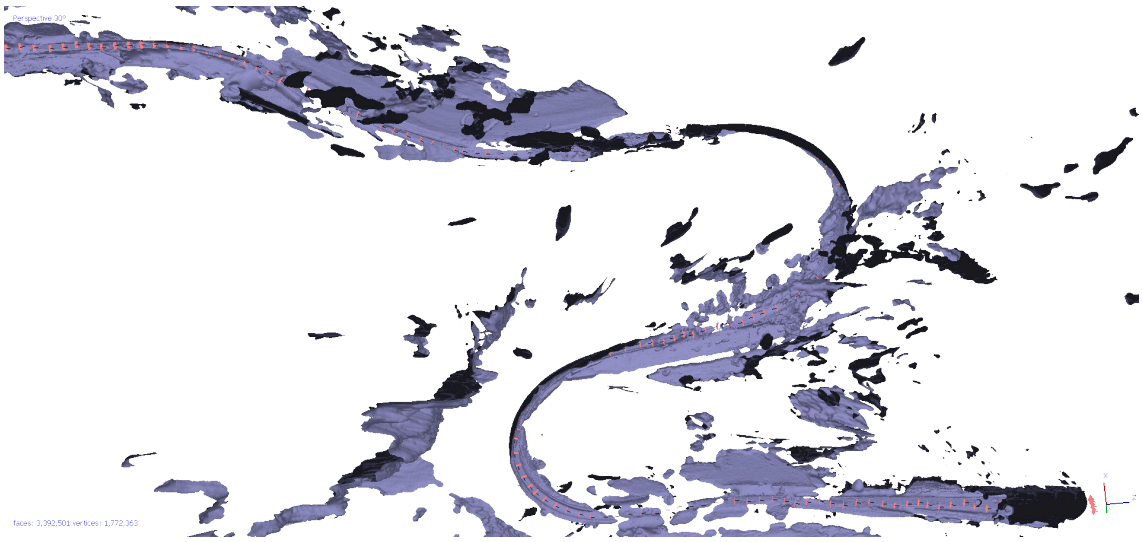
11.3 Automatic Track Surface Extraction and 3D Modelling

The bundle-adjusted combined dataset $IglsConcCN_{cba}$ was used for the generation of a reference 3D mesh for the track, which was manually cleaned around the edges of the track's surface. Instead of using one of the constrained bundle adjustment solutions, $IglsIce26C_{sim}$ was used in order to assess the result of the method on ice against the reference mesh and take into account the existence of distortions in the model, as well. The photogrammetric model of the SfM solution for $IglsConcC5_{arb}$, in an arbitrary reference system, was used to also test the transformation optimisation scheme, q.v. section 8.1.2. All meshes were created in PhotoScan, from the dense point cloud that was produced using the images at 1/64, 1/16, and 1/16 of the image resolution, respectively, and aggressive filtering without using masks.

Table 11.3 summarises the arc radii and the maximum vertical drop of the generated clothoid spline and 3D alignment, respectively. The success rate of cross section template detection, which was the process that generated the input to the alignment extraction process, is also reported. For $IglsIce26C_{sim}$, the radius estimation for the first bend was unreliable due to its short arc length which, in combination with the low detection success rates and despite reducing the extraction interval, led to the incorrect reconstruction of the bend's surface model, as an incorrect characteristic cross sectional curve was selected, q.v. figure 11.21 (upper left). A value of 25 m for D^{cloth} worked well for both $IglsConcCN_{cba}$ and $IglsConcC5_{arb}$. However, an additional control polygon vertex was returned between bends 9 and 10 for $IglsIce26C_{sim}$, as suppressing the noise in the detected template centre locations that were extracted from the mesh of the track in the iced state was not effective. Therefore, the value for this parameter was increased to 30 m in order to yield a clothoid spline with the correct number of bends.



(a) The visibility-consistent mesh was reconstructed directly from depths maps using the images at 1/64 of the original resolution with mild filtering and a single bounding box, cf. figure 11.16b. The camera stations are illustrated in red; large parts of the track's surface were not reconstructed.



(b) The mesh was reconstructed from a point cloud that was generated using the images at 1/64 of the original resolution with aggressive filtering and, cf. figure 11.16a.

Figure 11.16: Meshes of the first five bends of the Igls track that were generated from the $IglsIce26C_{sim}$ dataset in PhotoScan, using two reconstruction methods and a single bounding box.

Table 11.3: The arc radii and maximum vertical drop of the Igls track alignment as they were estimated from each reconstruction, as well as the success rate of the cross section template detection.

Dataset ID						Maximum Vertical Drop (m)
$IglsConcCN_{cba}$	Bend ID: Radius (m)					98.3
D^{cloth} = 25 m	K1: 20.3	K2: 16.5	K3: 17.0	K4: 59.3	K5: 52.8	
σ_n = 0.1 m	K6: 47.5	K7: 19.8	K8: 36.8	K9: 41.1	K10: 24.4	
	K11: 81.5	K12: 79.2	K13: 86.4	K14: 36.6	K15: 36.6	
	Template Detection Success Rate (extraction interval: 1 m)					
	left: 38%		full: 20%		right: 30%	
$IglsIce26C_{sim}$	Bend ID: Radius (m)					98.1
D^{cloth} = 30 m	K1: 29.2	K2: 16.4	K3: 17.1	K4: 61.9	K5: 51.3	
σ_n = 0.1 m	K6: 43.7	K7: 19.7	K8: 28.3	K9: 40.1	K10: 24.0	
	K11: 81.6	K12: 82.9	K13: 81.0	K14: 32.4	K15: 35.8	
	Template Detection Success Rate (extraction interval: 0.25 m)					
	left: 29%		full: 6%		right: 14%	
$IglsIce26C_{sim}$	Bend ID: Radius (m)					98.1
D^{cloth} = 30 m	K1: 55.5	K2: 16.6	K3: 17.1	K4: 60.7	K5: 47.3	
σ_n = 0.1 m	K6: 36.1	K7: 19.8	K8: 29.0	K9: 40.0	K10: 23.9	
	K11: 66.7	K12: 77.1	K13: 116.5	K14: 32.9	K15: 36.3	
	Template Detection Success Rate (extraction interval: 1 m)					
	left: 29%		full: 6%		right: 15%	
$IglsConcC5_{arb}$	Bend ID: Radius (m)					95.9
D^{cloth} = 25 m	K1: 17.4	K2: 15.2	K3: 17.8	K4: 57.7	K5: 52.9	
σ_n = 0.1 m	K6: 49.6	K7: 19.5	K8: 36.8	K9: 40.8	K10: 23.4	
	K11: 81.6	K12: 88.3	K13: 84.8	K14: 32.0	K15: 32.9	
	Template Detection Success Rate (extraction interval: 1 m)					
	left: 39%		full: 15%		right: 27%	

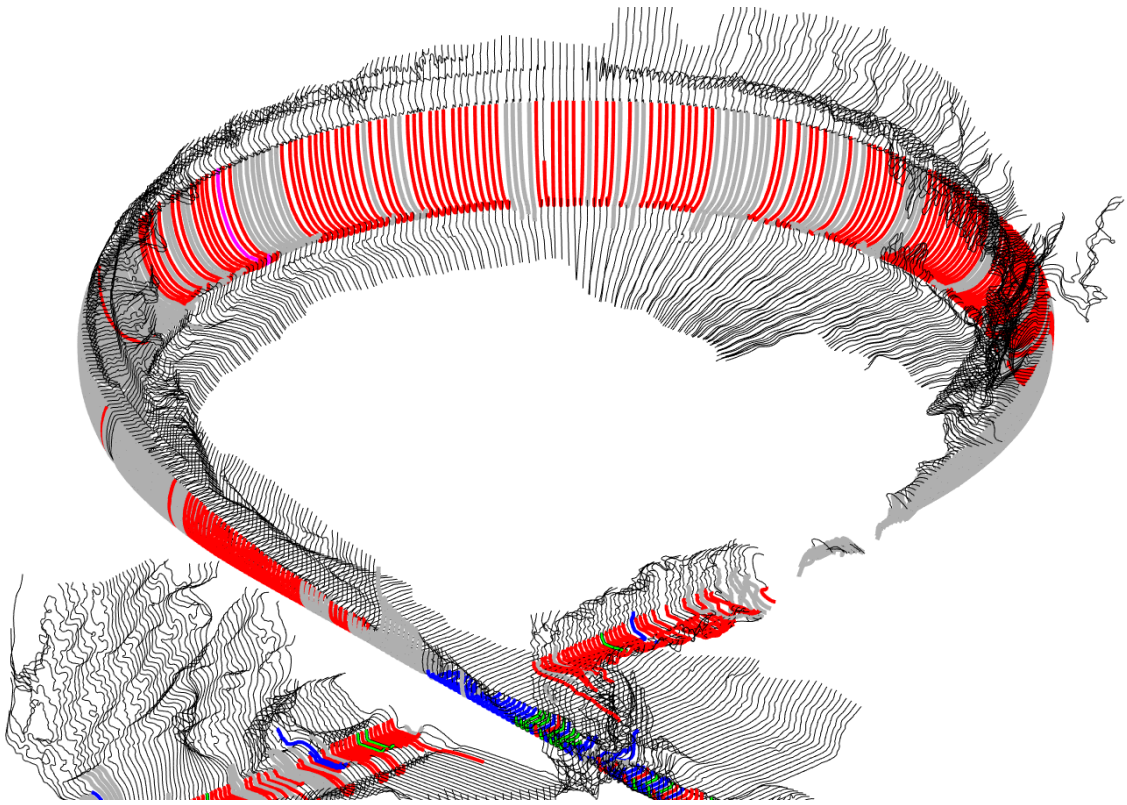


Figure 11.17: The characteristic cross section selection (magenta) at bend 7—the kreisel—of the Igls track for dataset $IglsIce26C_{sim}$.

The precision of the sweeping-and-morphing algorithm was initially assessed on the $IglsConcCN_{cba}$ reference mesh. Figure 11.18 shows the track's surface—which was generated in Rhino 5 (2018) by lofting between the cross-sectional curves that were automatically extracted with the proposed workflow—colour-coded according to the closest signed distance from the reference mesh. The mean and standard deviation of the distances was 0.05 m and 0.11 m, respectively, and below 0.90 m. The corresponding values for $IglsIce26C_{sim}$ were -0.04 m, 0.37 m, and 1.6 m, q.v. figure 11.20. In both cases, the largest discrepancies occurred along the transitions from the straight segments to the bend arcs. For the latter, the deviations were greater along the arc of the kreisel bend, as the characteristic cross section was selected slightly off-centre, q.v. figure 11.17. Finally, the track's surface model was generated from dataset $IglsConcC5_{arb}$ with its scale and orientation being estimated without using any reference information, according to the optimisation process that was described in section 8.1.2 and an initial approximation of 100 m for the maximum vertical drop, q.v. table 2.1. The mean and standard deviation of the signed distances was -0.45 m and 1.34 m, respectively. The maximum absolute distance was 3.9 m and the estimated maximum vertical drop was 95.9 m.

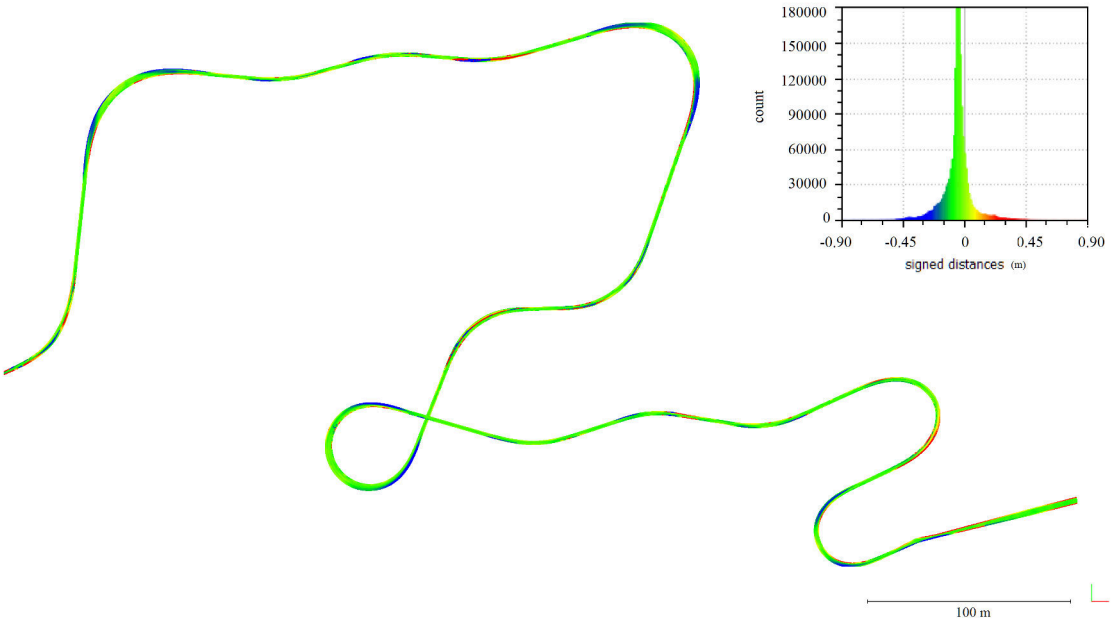


Figure 11.18: The Igls track surface model, which was generated from dataset $IglsConcCN_{cba}$, colour-coded according to its signed distance from the reference mesh.

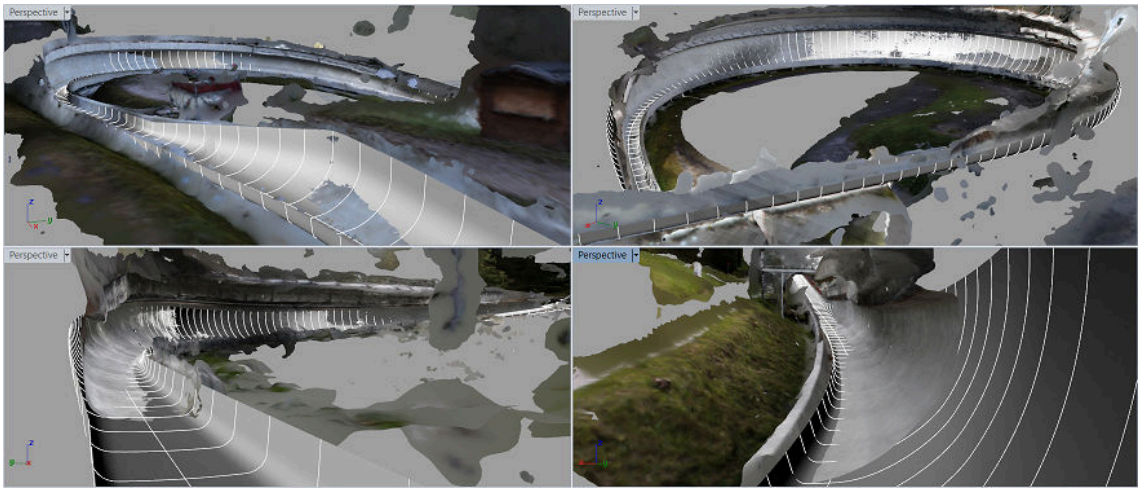


Figure 11.19: Views of the reconstructed 3D mesh and the extracted surface of the Igls (concrete) track for dataset $IglsConcCN_{cba}$ (reference).

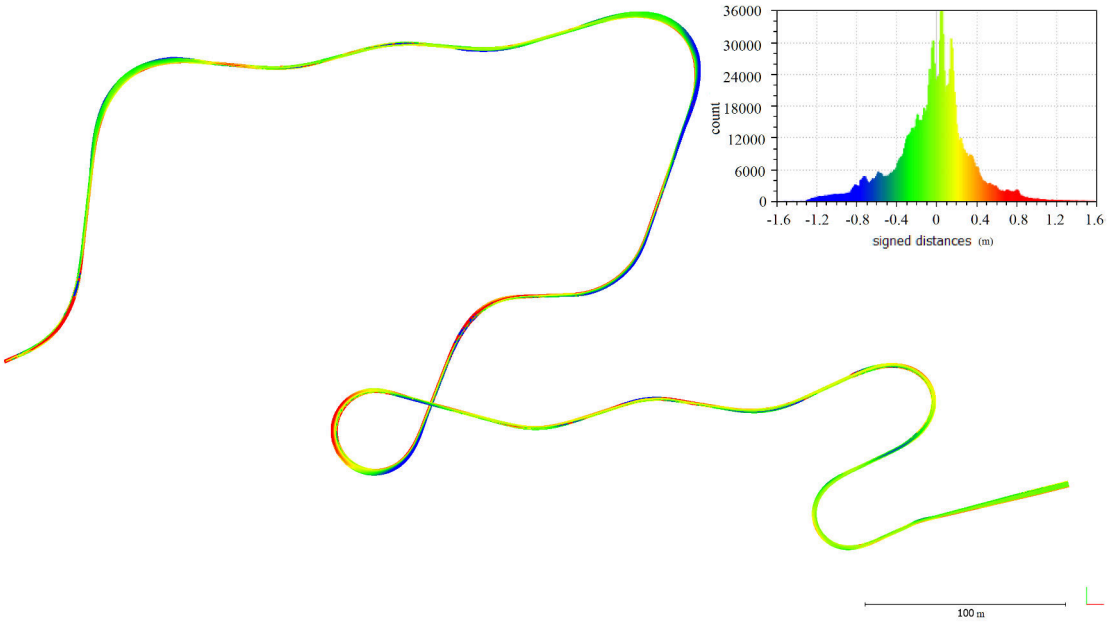


Figure 11.20: The Igls track surface model, which was generated from dataset $IglsIce26C_{sim}$, colour-coded according to its signed distance from the reference mesh $IglsConcCN_{cba}$.

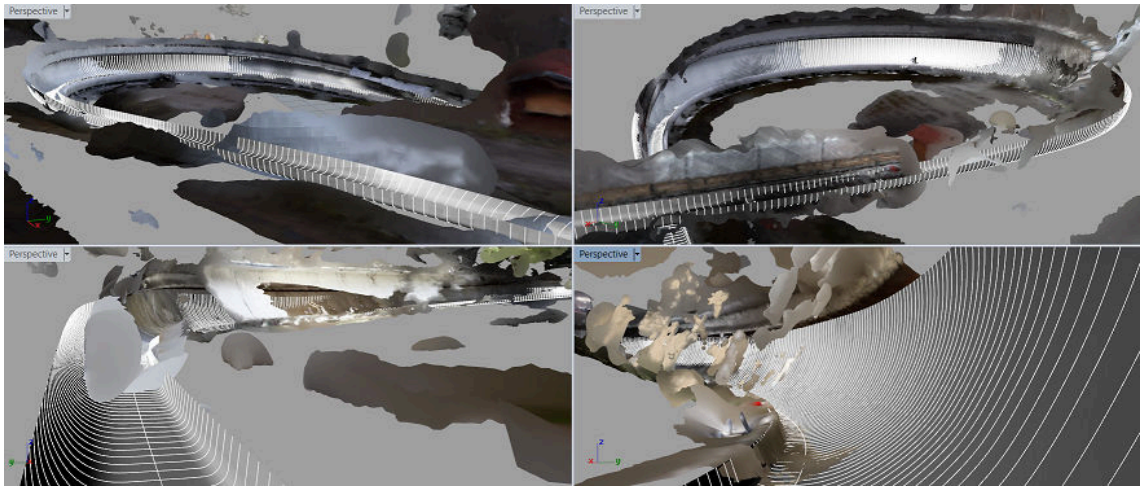


Figure 11.21: Views of the reconstructed 3D mesh and the extracted surface model of the Igls (ice) track for dataset $IglsIce26C_{sim}$.

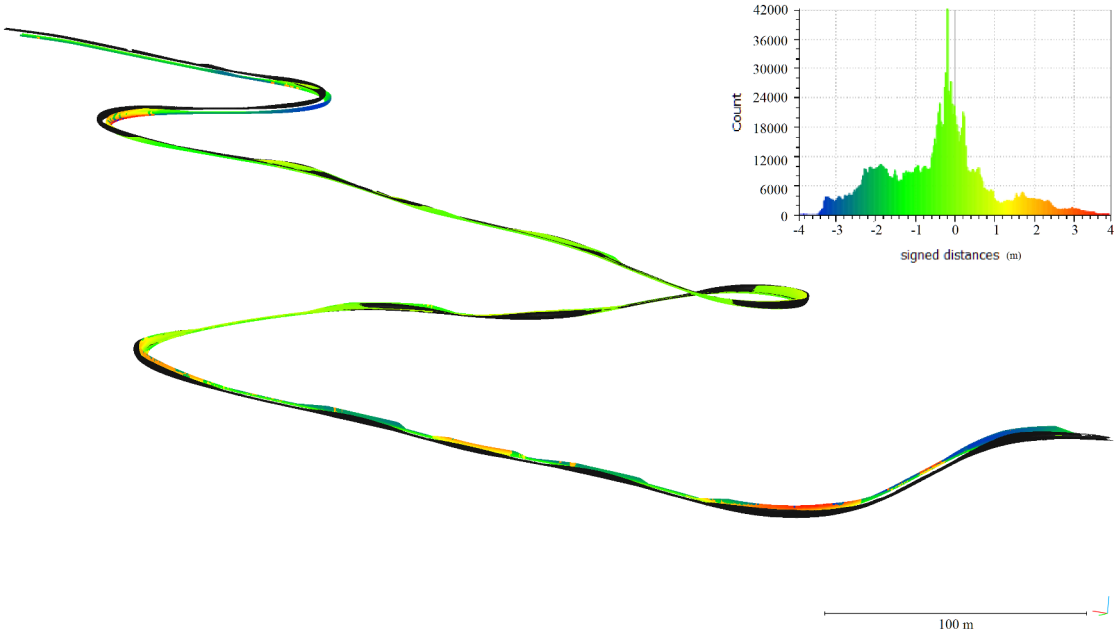


Figure 11.22: The Igls track surface model, which was generated from dataset $IglsConcC5_{arb}$, colour-coded according to its signed distance from the reference mesh $IglsConcCN_{cba}$ (black). Its scale and orientation was estimated without using any reference information.

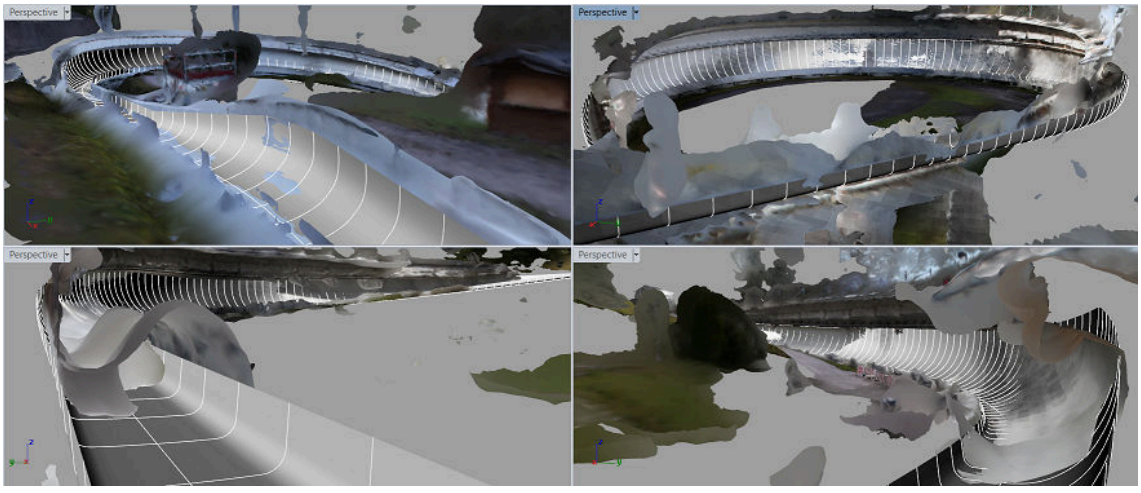


Figure 11.23: Views of the reconstructed 3D mesh and the extracted surface model of the Igls (concrete) track for dataset $IglsConcC5_{arb}$.

Chapter Summary

The Igls track scanning case study was primarily concerned with the geometric processing of the image-based reconstruction of the track as the image orientation process was successful—even though not always consistent, due to the non-deterministic nature of the process—for both the concrete and iced state. For the concrete state, 12 MP and 21 MP resolution images—acquired at an average pace of 2 images every 3.75 m and 1 image every 1.45 m, respectively—were collected within 77 min and 105 min, respectively. For the iced state, a dataset of 555 21 MP images was collected within 60 min, which corresponds to one image at every step using, on average, 2.75 m intervals. The success of this trial was primarily attributed to the fact that, for the most part, the Igls track was not covered by roofs and blinds. Nevertheless, the image acquisition intervals should be appropriately reduced during transitions between spaces with significantly difference lighting conditions. Despite the fact that the results of the preprocessing image segmentation could not be reliably propagated to the final 3D model, the proposed automatic track surface generation algorithm returned accurate and reliable results for the concrete state, especially when the reconstruction was based on GCPs that were measured at shorter than 150 m intervals along the track. For the iced state, extra care is required along bends with poor surface texture where the dense image matching is severely affected due to the reduced image overlap.

Part V

Epilogue

This final part brings together the entire thesis in order to discuss the findings that were reported in the previous chapters, with respect the objectives that were set out in section 1.3, and draw conclusions as to whether or not, and to what extent, the aim of the study, q.v. section 1.2, was fulfilled.

Chapter 12

Discussion

A set of methods pertaining image-based 3D scanning and as-built modelling were developed and assessed in the context of reconstructing a sliding track with minimal user interaction. The intent of objective №1, q.v. section 1.3, was to present relevant case studies so that the quality of any track scanning survey that lacks a rigorous assessment of the results—due to constraints with respect to access, time, cost, or complexity—and appears to be precise even though inaccurate, is not overestimated. To this end, the reference dataset that was created for the simulation experiment of a track bend indicated that a mildly convergent 3×3 image block at 1.5 m intervals along an ice track can completely reconstruct its surface. In addition, the results from 4 datasets that were collected along the entire length of 2 tracks, namely Lillehammer and Igls, were presented, q.v. chapters 10 and 11. The generated geometric elements were assessed against reference datasets of higher quality and between themselves; they were also compared with simpler approaches, in a attempt to verify that additional effort is not made in vain, q.v. section 10.3. Even the trials that failed to yield the desired result were informative of the limitations of the processes and the challenges that should be anticipated in future attempts. Depending on the availability of ground control, geometric accuracy better than 1 m (2-sigma error) was achieved for the image-based sparse reconstruction of a track in both the concrete and iced state. This means that, without considering the quality of the automatically generated parametric 3D model and depending on the survey control network density, the proposed photogrammetric approach can yield point clouds of comparable geometric quality to a terrestrial laser scanning survey within a fraction of time for on-site data acquisition, q.v. sections 2.3 and 11.2.

A set of hand-held laser scanning surveys was also conducted. In theory, an accuracy of 0.1 m can be achieved indoors with a walking pace of 500 m/h to 2800 m/h, q.v. section 3.6.2. However, the results that were obtained for Zeb1 datasets $Zeb1s1$ and $Zeb1s3$ suggested that if the track was

to be segmented into 250–500 m segments in order to keep the scan time low, accuracy of 0.6 m/km could be achievable. Assuming an average walking pace of 2700 m/h and a 50 m overlap between successive segments, scanning a 1450 m long track split in 7×250 m long segments would require 72 minutes of unobstructed track access; scanning a 1400 m long track split in 3×500 m long segments would require 56 minutes of unobstructed track access. Arguably, such an estimation is quite optimistic as it ignores registration errors between successive segments. After all, the reported RMSE were based on an optimal registration of each alignment to the reference curve which means that they may be underestimated for shorter ones.

The track geometry that was acquired with the Zeb1 was locally consistent and reliable with very few gaps but may suffer from nonlinear distortions—which, by design, cannot benefit from ground control measurements—and lacks intensity or colour information, q.v. section 3.6.2 and 10.3. The faster acquisition speed of the latest generation of the Zeb scanner may be more efficient at the same pace but closing a greater number of shorter loops, in order to keep the total scan time low, will be necessary and result in an increased aggregate survey duration. Returning from inside the track, which could potentially reduce the total scan time, defeats the purpose of the loop closure due to the working principle of the sensor and may also yield an inconsistent point cloud—in practice, this problem can be solved by keeping only the points collected while the sensor was on the track. However, as the view or access towards the inside of the track may be blocked, closing short loops may not be possible. Alternatively, a longer path around the track or through the forest area, where plenty of geometric features are available, could be selected for the loop closure but this would increase the length and duration of the scan.

With respect to image-based scanning, the on-site data collection strategy and its effect on the outcome was investigated towards objective №2, as dense reconstruction with a forward-facing camera and extreme depth variation within the field of view is known to be suboptimal and is primarily avoided—unless, the acquisition speed is as important as the accuracy of the outcome. The computer simulations for photogrammetric network design were based on images of real and artificial scenes with high ground resolution textures, without attempting to replicate the surfaces' reflectance properties, in contrast to what was suggested in Slocum et al. (2017). A set of generic imaging configurations were considered, based on the premise that under favourable conditions it would be straightforward to isolate and identify the effect of the imaging network geometry on the reconstruction accuracy. As the image point correspondences cannot be taken for granted in scenarios that involve oblique imaging of non-signalised scenes—in contrast to the common practice that involves coded targets and to approaches where the scene was represented by a sparse

3D point array from a lower resolution reconstruction (Alsadik et al., 2013)—the feature matching capabilities of the photogrammetric process were taken into account, as well.

It was shown that the more complicated imaging geometries are not necessarily better and confirmed that the simplest—i.e. linear camera motion—are indeed the worst even compared to comparably weak configurations. However, with respect to the determination of a single optimal imaging configuration, the results were deemed inconclusive due to the competing nature of small versus large convergence angles for 3D reconstruction and image matching. A computer generated network resulting from a numerical optimisation scheme such as those proposed in Oliensis (2001), Saadatseresht et al. (2005), or Alsadik et al. (2013) could yield a theoretically better solution but it would be impractical to precisely implement it in this setting. Rather conservative along-track intervals where considered for the simulations, in order for the relative image orientation of all the images for all the assessed variations to be retrieved, so that comparable results could be generated. In practice, the interval length is one of the most decisive factors that defines the effective overlap between successive images, especially for fully covered tracks. However, in this setting, it is primarily dictated by the available resources and allocated time on the track. Only high resolution cameras where considered but the assessed imaging configurations can be replicated within a videogrammetric workflow. The increased blur effects due to the lateral camera motion could be mitigated by slower forward motion; however, this could potentially lead to a large number of lower resolution frames acquired within the same amount of time it would take a DSLR camera to cover the same scene.

Despite the fact that the image orientation within PhotoScan (2018) is an automatic process, at several occasions, user intervention was required in order for it to be successful. This effect was primarily attributed to the imaging configuration comprising forward-facing cameras, but not as a failure of the affine invariant feature matching process; in fact PhotoScan was pretty robust in finding correspondences in feature space and would only struggle with limiting cases of non-Lambertian surfaces or significantly low image overlap combined with vastly different exposure levels due to inconsistent lighting conditions. Even though it is more effective in processing images acquired by a forward-facing camera compared to VisualSfM (Wu, 2011), for example, it appears that its matching strategy is tailored towards image blocks of more conventional configurations, e.g. aerial or strongly convergent blocks. It is presumed that the iterative local bundle adjustments within the SfM scheme, which is used in order to purge false matches, is not appropriate for the orientation of images that were acquired sequentially along a path, even if approximate values for the exterior orientation that reflected this configuration were provided, as the automatic process

would not retrieve a complete network if all the images are processed at once. This problem was solved by incrementally estimating the relative orientation of successive image subsets—automating this process requires the image set to be properly ordered. The problems with the orientation, at different image sampling levels, of the sequences that were collected along the Igls track were attributed to the different feature match acceptance thresholds that are implicitly applied by PhotoScan and to the relative range of image scales that are used for feature detection.

With regard to the problematic image sequences along the Lillehammer and Winterberg tracks, the potential of using an online feedback loop, in order to inform the user about the potential success or failure in estimating the relative orientation of each image with respect to the previously captured set without actually running the matching process, was considered towards fulfilling objective №6. The intent was to identify whether there was a clear correlation between the success or failure of the image matching and alignment process and certain quantifiable image quality metrics which can be quickly calculated online. The effect of the image quality on the overall accuracy of the reconstruction was not considered.

Due to practical reasons, as the workflow under consideration involves a hand-held DSLR camera, a mobile phone is more appropriate—compared to a laptop—as a processing device for the provision of feedback on the quality of every image. However, the computational capabilities of a portable device are not be adequate for high resolution image processing, yet. Another problem stems from the fact that the exact details of the detector-descriptor combination and matching algorithm may be unknown if a proprietary software is used for offline processing. In addition, due to the non-deterministic nature of feature detection and matching, different runs of the process on the same set of images using the same settings may fail to align different image pairs each time. If the processing is to be conducted on a remote server, the overhead of transferring the data from the camera should also be considered.

On the other hand, the usage of a hand-held camera means that the operator is in control and in a position to guide the process. Even though the image quality estimation in table 10.3 was subjective, there is no indication that an image quality metric can be objective without a reference image, q.v. section 3.2.2, or that the control of the quality assurance of the acquired images can be taken away from the camera operator in this application setting. If such an online feedback scheme was to be used in practice, in the best case scenario—where the feedback for each and every image is positive the first time around, allowing the user to move on to the next station—at least a couple of seconds per image may be added to the duration of the data collection, q.v. section 3.2.1. The worst case scenario would have the user caught up in an infinite loop of failed attempts.

For example, the failed attempt to reconstruct the entire Lillehammer track was attributed to the combination of intervals longer than 4 m and glossy ice surfaces, both of which can be handled better by the operator alone without an additional feedback loop. Even if there is additional time to actually run the image matching process for each acquired image, it might be better spent on acquiring more images.

The most challenging task towards objective №3—with regard to automating the 3D modelling of the track from a point cloud or mesh with noise, outliers, and gaps—was extracting the individual geometric components from the scene's point cloud; the most important one being the track's surface. Isolating each object of interest was essential in compensating for missing information and in removing extraneous parts, so that their geometric properties could be modelled with any one of the common least squares frameworks for parametric surface approximation. Subsequently, the appearance attributes of each individual component could also be manipulated.

Segmenting the track in image space was considered in pursue of objective №5. The interactive "Lazy Snapping" segmentation algorithm was automated by integrating geometric information relevant to the specific task. However, even in cases where the track's edges are sharply segmented from the background, there is no guarantee that this result will be propagated in 3D space through the dense image matching, point cloud generation, and mesh reconstruction stages. Nevertheless, the process was beneficial for the detection of the track's start line. It is clear that this is a trivial task for a human operator but enhancing the track model with this attribute within an automated track modelling process that feeds a simulation system is essential. It is worth noting, however, that the start line detection on the images of the concrete track was not feasible, as the brute-force matching method could not cope with the amount of candidates that were generated. The exact location of the finish line is not equally important. As the primary aim of the simulation is to improve the performance between training runs and not predict the total run time, a virtual finish line can be located at certain distance from the start line, according to the total competition length, or somewhere along the deceleration stretch after the last bend.

Both of these lines can be defined with respect to a curve that represents the track's alignment. Fitting a generic spline to the sensor's trajectory can be used for this purpose but a geometric curve model that captures the track's design intent is more appropriate—in fact, one of the shortcomings of the proposed workflow is that a generic spline curve was used for the extraction of the alignment's vertical profile. Objective №4 was fulfilled by developing a clothoid spline fitting algorithm for the extraction of the track's horizontal alignment, which can also be applied to as-built modelling of roadway infrastructure. The problems that were identified in the literature, with

respect to the usage of a control polygon for the parameterisation and the detection of compound bends and arcs with very low curvature (Holgado-Barco et al., 2014; Nguyen et al., 2014; Bauer et al., 2009; Garach et al., 2014), were partially overcome. Experiments showed that a representative primitive type sequence can be extracted, over-segmentation can be kept to a minimum, and the results are not too sensitive to the required user-defined parameters.

In some of the existing approaches for clothoid spline fitting (Nguyen et al., 2014; Baran et al., 2010), local control was imposed by over-segmentation; this was considered a desirable side-effect of avoiding the curvature-based curve segmentation process. Seeking the control polygon of the alignment may lead to computational instabilities but it provides a CAD-user with a convenient way to make local adjustments and identify the appropriate user action that will fix potential problems. Unlike previous work, clothoid segments are taken into consideration and not treated as nuisance parameters, which makes the estimation of reliable initial approximations for all the parameter values even more challenging. Even though Cornucopia (2010) was the fastest implementation for clothoid spline fitting, any gain in time is lost in the trial-and-error process for tuning all the parameters involved, in order to get the best possible result, which may not be acceptable; admittedly, it was not designed for the extraction of road centreline alignments.

One might argue that a control polygon can be constructed as a by-product on top of a graph-based solution. However, one of the reasons for using it, apart from guiding the curve fitting process, is controlling the total number of vertices, which in turn defines the total number of track bends. In the general case, where no reference data is available, the total number of detected bends, the maximum vertical drop, the percentage of the cross sections where a template was detected, and the track width and floor orientation—as they were estimated on each detected cross section template—can provide a strong indication of the 3D model's structural integrity for the quality assessment of the reconstruction result.

The proposed surface modelling technique, q.v. section 8.3.3, combines many of the qualities that were described in Wu et al. (2012), Yin et al. (2014), and Dimitrov et al. (2016) but also compensates for cluttered point clouds of surfaces with boundaries, without the need for user interaction. This automatic sweeping-and-morphing algorithm generates a continuous 3D model of the track's surface; therefore, data points can be sampled at any desired resolution and used as an input to the physics engine of a simulating system, q.v. section 2.2.2. The aspect of the completeness and density of the point cloud, which was generated from an image-based or laser scanning survey, and potential problems related to it, was investigated in sections 10.4 and 11.3. The main challenge, especially with respect to meshes with large gaps and geometric artifacts, was

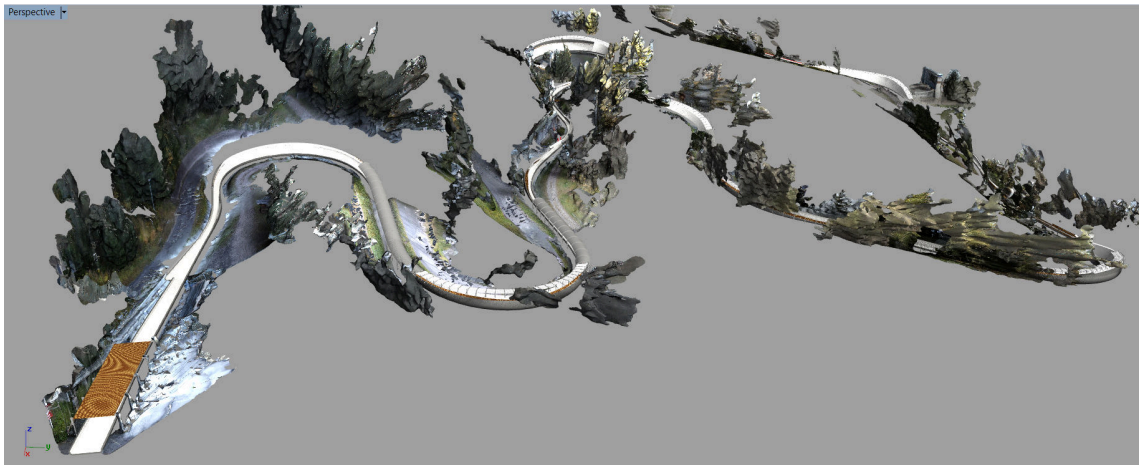


Figure 12.1: The Igls track model. The 3D mesh was generated in PhotoScan (2018) from the image-based scanning survey of the track's concrete structure. Texture maps were generated from the collected images and applied to the surrounding environment. The track's surface was extracted by the proposed workflow, manually refined, and used to guide the modelling of the blinds, roofs, etc. Basic settings were used for the materials, textures, and lighting for this illustration.

selecting the characteristic cross sections for the interpolation. If not enough cross sections exist, generic templates can be used instead; even though this might turn out to be a poor choice along the bends, it could avoid a severely distorted surface model, q.v. figure 11.21.

A complete assessment of whether or not the accuracy of the resulting track geometry is sufficient for the simulation of a sled run cannot be separated from the simulating system, as it depends on its sensitivity. Even though the effectiveness of a system, given a 3D model of the track, relies on whether or not it can realistically simulate the sled's motion and present an immersive experience to the athlete (Biocca et al., 1995, p. 57; Lombard et al., 1997, p. 6), such a comprehensive sensitivity analysis was beyond the scope of this study, which demonstrated the potential of each scanning method in order to be taken into account for the design of such systems and during the interpretation of their performance. Some of the results that were produced by this work were used by one, but the techniques were not developed in tandem with—or specifically for—a particular virtual reality system in mind, as there can be a wide variety of component combinations with varying specifications and requirements. Nonetheless, the fact that the proposed method's results were beneficial to the sport's community was confirmed by anecdotal positive feedback from the assessment of the Igls track model, q.v. figure 12.1 and 12.2, which was based on empirical evidence.

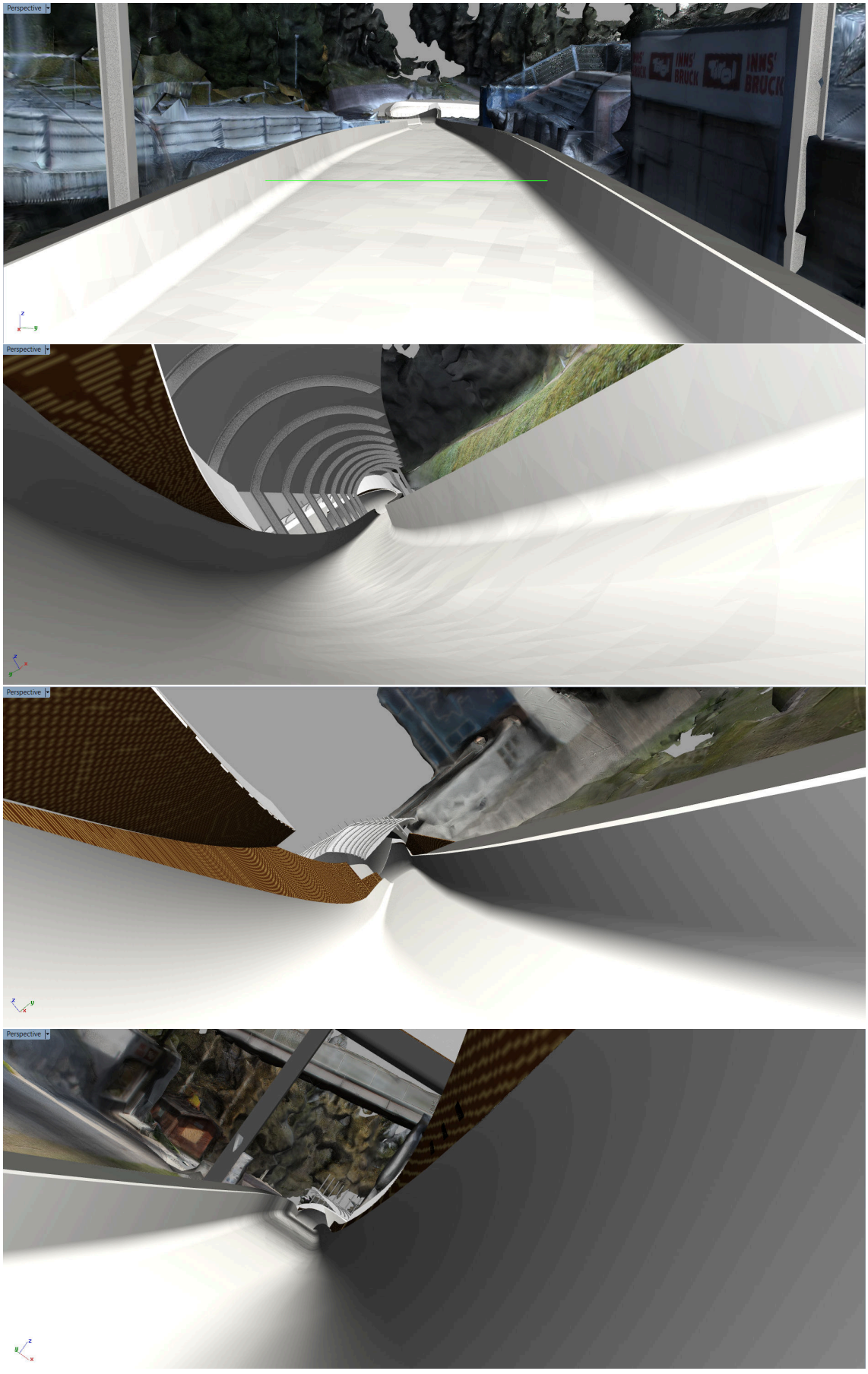


Figure 12.2: Views of the Igls track model, q.v. figure 12.1.

Chapter 13

Conclusions

Even in the highly unlikely event that the data collected during a photogrammetric or laser scanning survey as well as the reconstructed geometry are flawless, the entire process of creating the 3D model of a track for an immersive virtual environment for sliding simulations cannot be carried out by a non-specialist alone, yet. Nevertheless, a substantial part of the manual effort in generating the most important parts of the scene, namely the track's alignment and surface, can be automated with low-cost software and hardware, at least to the extent where a reliable initial approximation is acquired.

The 3D reconstruction and an approximation of the 3D model of the track's surface can be retrieved within 24 hours, depending on the available processing power, without user interaction. In order to achieve full surface coverage with an image-based point cloud, all the effort must be focused on a small track segment, e.g. completely scanning a 150 m segment in the iced state would require up to 1 hour on site and 250–500 images. Such an approach would be appropriate in order to constantly update a segment of an existing base model of a track. The image-based 3D scanning of the entire length of a 1.5 km long sliding track can be successfully completed within less than 2 hours, without a requirement for scene signalisation, by a single person with a hand-held, high-resolution, non-metric, pre-calibrated DSLR camera with a moderately wide prime lens. Moving the camera in a linear motion along the track must be avoided. The along-track intervals for image acquisition must be kept strictly under 4 m, even when the concrete structure of a sliding track is scanned. In the vicinity of parts of the track that are covered by roofs, blinds, or glossy ice, the intervals should be reduced to 1–2 m. This judgement is best left to the camera operator, assuming familiarity with standard image acquisition and composition practices. If there is a restriction on the number of images that can be collected within the allocated time, the number of images per station should be reduced instead of increasing the interval; if necessary, a second

camera should be used in order to divide the workload the track in half.

The relative orientation of the image set should be estimated incrementally in sequential subsets comprising no more than 100–200 images; this can be achieved automatically without manual image observations by low-cost software. Agisoft (PhotoScan, 2018) is a popular commercial option that has been successfully applied on problems with similar accuracy expectations; nevertheless, the integrity of the result and the repeatability of the process are not guaranteed. Subdecimetre accuracy can be achieved for the track's sparse reconstruction if ground control points are measured, with the typical GNSS RTK technique accuracy at 100–150 m intervals along the track, and used in a constrained self-calibrating bundle adjustment. If only 3 ground control points are available, submetre accuracy is achievable, even if they are not used in the bundle adjustment and only provide for the similarity transformation with respect to a local coordinate system. If a fully automated processing workflow is required but no reference information with submetre accuracy is available, the photogrammetric model's datum can be defined from existing knowledge, either in the form of an approximate 3D alignment extracted from an orthoimage and a digital surface model of the track site or from information about the geometric structure of the track itself. Such a model should only be used for virtual track walks but not for sliding simulations.

The alternative option of a hand-held laser scanner, i.e. the Zeb1, is more expensive but can yield a dense 3D point of the entire track's surface with very few gaps and outliers, which can significantly simplify the 3D modelling process. However, even though the range measurements and integrated IMU directly solve the datum definition problem, colour or texture information would have to be added to the point cloud or mesh, respectively, using an additional image-based workflow. Despite the fact that the data acquisition rate is faster, and even greater for the latest sensor generation, i.e. ZEB-REVO, the limited outdoor range and the proportionality of the positional drift to the operation time renders the total survey duration for a 1.5 km long track comparable to what is required in order to achieve similar results with the photogrammetric technique.

The automatic identification of individual features in a track scene, such as roofs and blinds—as well as the extents of the track itself—in order to realistically represent their geometric and visual attributes within a virtual environment, should be performed in 3D object-space and ideally take into account their image projections, as well. Mesh reconstruction algorithms which do not consider image-space information may not retain the edges that were detected on the images—even if they were used to reduce the generated point cloud—and those that do, are primarily suitable for objects that are visible in their entirety on each image. Therefore, segmenting the track's projection on the images as a pre-processing step should only be considered within a fully

automated process, in order to facilitate the detection of the track's start line or in order to better control the size of the generated point cloud.

Fitting a clothoid spline to a set of 2D points without user interaction requires significant computational effort but can effectively compensate for missing parts, along and across the track, and yield a smooth model, if compensated by an accurate vertical profile. The proposed sweeping-and-morphing algorithm can provide an initial solution for the parametric 3D model of the track's surface, without requiring user interaction, but the result needs refining, particularly around the edges and along the transitions in and out of the bends. Specifically for a track that was scanned in the iced state, and especially if the model is to be used in a virtual reality system, visual inspection and sliding simulations for the verification of the result are required.

Bibliography

Aanes, H., et al. (2012). Interesting Interest Points. *International Journal of Computer Vision*, 97(1):18–35.

Achanta, R., et al. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 34(11):2274–2282.

Aggarwal, C. (2015). *Data Mining*. Springer.

Albertz, J. (2007). A look back, 140 Years of Photogrammetry. *Photogrammetric Engineering & Remote Sensing*, page 505.

Alsadik, B. (2014). *Guided Close Range Photogrammetry for 3D Modelling of Cultural Heritage Sites*. PhD thesis.

Alsadik, B., et al. (2013). Automated camera network design for 3D modeling of cultural heritage objects. *Journal of Cultural Heritage*, 14(6):515–526.

Alsadik, B., et al. (2014). Minimal camera networks for 3D image based modeling of cultural heritage objects. *Sensors (Basel, Switzerland)*, 14(4):5785–804.

Altenberg (2016). Altenberg bobsleigh, luge, and skeleton track. *http://www.wia-altenberg.de/bobbahn-altenberg/bahninfo.php [Accessed on 01/09/2017]*.

An, Y., et al. (2011). Geometric properties estimation from discrete curves using discrete derivatives. *Computers & Graphics*, 35:916–930.

Apollonio, F. I., et al. (2014). Evaluation of feature-based methods for automated network orientation. *ISPRS Archives*, XL-5:47–54.

Arthur, D., et al. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

Atkinson, K. B. (1996). *Close range photogrammetry and machine vision*. Whittles Publ.

Australis (2007). Photometrix Australis Manual, www.photometrix.com.au.

AutoCAD Civil 3D (2018). *https://www.autodesk.co.uk/products/autocad-civil-3d/overview*. Autodesk, Inc., San Rafael, CA USA.

Autodesk 123D Catch (2009). [discontinued: redirects to https://www.tinkercad.com/]. *http://www.123Dapp.com/catch*.

Autodesk 3ds Max (2015). *https://www.autodesk.co.uk/products/3ds-max/overview*. Autodesk, Inc., San Rafael, CA USA.

Baarda, W. (1973). S-Transformations and Criterion Matrices. *Publications on Geodesy*, 5(I).

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122.

Baltsavias, E. P. (1999). A comparison between photogrammetry and laser scanning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3):83–94.

Baran, I., et al. (2010). Sketching Clothoid Splines Using Shortest Paths. *Computer Graphics Forum*, 29(2):655–664.

Barazzetti, L. (2011). *Automatic Tie Point Extraction from Markerless Image Blocks in Close-Range Photogrammetry*. PhD thesis.

Barazzetti, L. (2017). Network Design in Close-Range Photogrammetry with Short Baseline Images. *ISPRS Annals*, pages 17–23.

Barentzen, J. A., et al. (2012). *Guide to Computation Geometry Processing*. Springer US, London.

Bartoň, M., et al. (2014). Detection and reconstruction of freeform sweeps. *Computer Graphics Forum*, 33(2):23–32.

Bassani, M., et al. (2016). Identification of Horizontal Circular Arc from Spatial Data Sources. *Journal of Surveying Engineering*, 142(4):1–14.

Bauer, U., et al. (2009). Generating parametric models of tubes from laser scans. *Computer Aided-Design*, 41(10):719–729.

Bay, H., et al. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.

Bell, S., et al. (2015). Material recognition in the wild with the materials in context database. In *IEEE CVPR*, pages 3479–3487.

Belyaev, A. (2004). Plane and space curves, curvature, curvature-based features. Technical report, Max-Planck Institut für Informatik.

Ben-Arieh, D., et al. (2004). Geometric modeling of highways using global positioning system data and B-spline approximation. *Journal of Transportation Engineering*, 130(5):632–636.

Berger, M., et al. (2016). A Survey of Surface Reconstruction from Point Clouds. *Computer Graphics Forum*, pages 1–29.

Bertolazzi, E., et al. (2010). Fast and Accurate Clothoid Fitting. *ACM Transactions on Mathematical Software*, V(212):30.

Bertolazzi, E., et al. (2014). G1 fitting with clothoids. *Mathematical Methods in the Applied Sciences*, 38(5):881–897.

Besl, P. J., et al. (1992). A Method for Registration of 3-D Shapes. *IEEE TPAMI*, 14(2):239–256.

Bhatla, A., et al. (2012). Evaluation of accuracy of as-built 3D modeling from photos taken by handheld digital cameras. *Automation in Construction*, 28:116–127.

Bhushan, B. (2013). *Introduction to tribology*. John Wiley & Sons.

Bideau, B., et al. (2010). Using Virtual Reality to Analyze Sports Performance. *IEEE Computer Graphics and Applications*, 30(2):14–21.

Bing Maps (2017). https://www.bing.com/maps. *Microsoft*.

Biocca, F., et al. (1995). Immersive virtual reality technology. *Communication in the Age of Virtual Reality*, pages 57–124.

Blais, F. (2004). Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1):231–243.

Blender (1995). Blender Foundation. *www.blender.org*.

BLOM AS (2017). *Geographic information provider, http://www.blomasa.com/.*

Blum, H. (1967). A transformation for extracting new descriptions of shape. In *Symposium on Models for the Perception of Speech and Visual Form*.

Bo, P., et al. (2016). A graph-based method for fitting planar B-spline curves with intersections. *Journal of Computational Design and Engineering*, 3:14–23.

Borlin, N., et al. (2016). External verification of the bundle adjustment in photogrammeric software using the damped bundle adjustment toolbox. *ISPRS Archives*, XLI-B5:7–14.

Bosché, F., et al. (2015). The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM : The case of cylindrical MEP components. *Automation in Construction*, 49:201–213.

Bosse, M., et al. (2008). Map matching and data association for large-scale two-dimensional laser scan-based SLAM. *International Journal of Robotics Research*, 27(6):667–691.

Bosse, M., et al. (2009). Continuous 3D Scan-Matching with a Spinning 2D Laser. In *IEEE International Conference on Robotics and Automation*, pages 4244–4251, Kobe, JAPAN.

Bosse, M., et al. (2013). Place Recognition Using Keypoint Voting in Large 3D Lidar Datasets. In *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany.

Bosse, M., et al. (2012). Zebedee: Design of a Spring-Mounted 3-D Range Sensor with Application to Mobile Mapping. *IEEE Transactions on Robotics*, 28(5):1104–1119.

Botsch, M., et al. (2010). *Polygon Mesh Processing*. CRC Press.

Boykov, Y. Y., et al. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *IEEE 8th ICCV*, volume 1, pages 105–112.

Bradley, D., et al. (2007). Adaptive thresholding using the integral image. *Journal of Graphics Tools*, 12(2):13–21.

Braghin, F., et al. (2011). Multi-body model of a bobsleigh: comparison with experimental data. *Multibody System Dynamics*, 25(2):185–201.

Braghin, F., et al. (2010). Design and Verification of Bobsleigh Track. *Proceedings of the ASME 10th Biennial Conference on Engineering Systems Design and Analysis*, 4:505–512.

Brey, P. (2008). Virtual reality and computer simulation. *The handbook of information and computer ethics*, page 361.

Brilakis, I., et al. (2008). Shape-Based Retrieval of Construction Site Photographs. *Journal of Computing in Civil Engineering*, 22(February):14–20.

Brilakis, I., et al. (2005). Material-Based Construction Site Image Retrieval. *Journal of Computing in Civil Engineering*, 19(October):341–355.

Brilakis, I., et al. (2015). Infrastructure computer vision. *Advanced Engineering Informatics*, 29(2):147–148.

Brilakis, I., et al. (2010). Toward automated generation of parametric BIMs based on hybrid video and laser scanning data. *Advanced Engineering Informatics*, 24(4):456–465.

Bromley, K. L. (1999). *Factors Affecting the Performance of Skeleton Bobsleigh*. PhD thesis, University of Nottingham.

Brostow, G. J., et al. (2008). Segmentation and recognition using structure from motion point clouds. In *ECCV*, pages 44–57. Springer.

Brown, D. C. (1958). A solution to the general problem of multiple station analytical stereotriangulation. Technical Report No. 43 (AFMTC TR 58-8), Patrick Airforce Base, Florida.

Brown, D. C. (1971). Close-Range Camera Calibration. *Photogrammetric Engineering*, 37(8):855–866.

Brown, D. C. (1980). *Application of close-range photogrammetry to measurements of structures in orbit*. Geodetic Services Incorporated.

Brown, M., et al. (2003). Advances in computational stereo. *IEEE TPAMI*, 25(8):993–1008.

Buchin, M., et al. (2011). Segmenting trajectories : A framework and algorithms using spatiotemporal criteria. *Journal of Spatial Information Science*, 3:33–63.

Bucksch, A. K. (2011). *Revealing the skeleton from imperfect point clouds*. Phd, Technische Universiteit Delft.

Budroni, A., et al. (2010). Automated 3D Reconstruction of Interiors from Point Clouds. *International Journal of Architectural Computing*, 08(01):55–74.

Bundler (2008). Snavely, Noah. *http://www.cs.cornell.edu/~snavely/bundler/*.

Calabi, E., et al. (1998). Differential and Numerically Invariant Signature Curves Applied to Object Recognition. *International Journal of Computer Vision*, 26(2):107–135.

Calonder, M., et al. (2010). BRIEF: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792.

Campbell, R. J., et al. (2001). A Survey Of Free-Form Object Representation and Recognition Techniques. *Computer Vision and Image Understanding*, 81(2):166–210.

Casal, G., et al. (2017). Optimization of horizontal alignment geometry in road design and reconstruction. *Transportation Research Part C: Emerging Technologies*, 74:261–274.

Castro, M., et al. (2006). Geometric modelling of highways using global positioning system (GPS) data and spline approximation. *Transportation Research Part C: Emerging Technologies*, 14(4):233–243.

Castro, M., et al. (2005). Alignment indices as a tool to evaluate safety and design consistency in two lane rural roads. In *Proceedings of the Third International Symposium on Highway Geometric Design*.

Cazals, F., et al. (2003). Estimating Differential Quantities using Polynomial fitting of Osculating Jets. Technical report, INRIA.

Charrier, C., et al. (2015). Comparison of no-reference image quality assessment machine learning-based algorithms on compressed images. In *SPIE Electronic Imaging*, number Proc. SPIE 9396.

Chen, Y., et al. (1992). Object Modeling by Registration of Multiple Range Images. *Image and Vision Computing*, 10(3):145–155.

Cheng, Y.-j., et al. (2016). Automatic Extraction of Tunnel Lining Cross-Sections from Terrestrial Laser Scanning Point Clouds. *Sensors*, 16(1648).

Chernov, N. (2010). *Circular and linear regression: Fitting circles and lines by least squares*. CRC Press.

Chernov, N. (2012). *http://people.cas.uab.edu/ mosya/cl/MATLABcircle.html* . MATLAB code for circle fitting algorithms.

Choi, B., et al. (1990). Sweep surfaces modelling via coordinate transformation and blending. *Computer-Aided Design*, 22(2):87–96.

Choi, S., et al. (2007). Data Generalization Algorithm for the Extraction of Road Horizontal Alignment Design Elements Using the GPS / INS Data. *Advances in Hybrid Information Technology, Lecture Notes in Computer Science*, 4413:51–62.

CIE (1976). COLORIMETRY — PART 4: CIE 1976 L*A*B* COLOUR SPACE, International Commission on Illumination. *ISO 11664-4:2008(E)/CIE S014-4/E:2007*.

Cignoni, P., et al. (2008). Meshlab: an open-source mesh processing tool.

Cleveland, W. S., et al. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610.

CloudCompare (2017). version 2.8. *GPL software, Retrieved from http://www.cloudcompare.org/*.

Cooper, M. A. R., et al. (1988). Statistical concepts and their application in photogrammetry and surveying. *The Photogrammetric Record*, 12(71):637–663.

Cordes, K., et al. (2015). Feature Evaluation with High-Resolution Images. In *Computer Analysis of Images and Patterns, Lecture Notes in Computer Science*, volume 9256, pages 374–386.

Cordes, K., et al. (2013). High-Resolution Feature Evaluation Benchmark. In *15th CAIP, Lecture Notes in Computer Science (LNCS)*, pages 327–334.

Corke, P., et al. (2007). An introduction to inertial and visual sensing.

Cornea, N. D., et al. (2007). *Curve-Skeleton Properties , Applications and Algorithms*, volume 13.

Cornucopia (2010). http://www.mit.edu/˜ibaran/curves/.

Craig, C. (2013). Understanding perception and action in sport: how can virtual reality technology help? *Sports Technology*, 6(4):161–169.

Cremean, L. B., et al. (2006). Model-based estimation of off-highway road geometry using single-axis LADAR and inertial sensing. *Proceedings - IEEE International Conference on Robotics and Automation*, 2006(May):1661–1666.

Crété-Roffet, F., et al. (2007). The blur effect: Perception and estimation with a new no-reference perceptual blur metric. In *SPIE Electronic Imaging Symposium Conf Human Vision and Electronic Imaging*, volume 12, pages EI—-6492.

Cross, P. A. (1983). Advanced Least Squares applied to position fixing. Technical Report 2223.

Czerniawski, T., et al. (2016). Pipe spool recognition in cluttered point clouds using a curvature-based shape descriptor. *Automation in Construction*, 71(Part 2):346–358.

Dainty, A., et al. (2015). Don't Believe the (BIM) Hype : The Unexpected Corollaries of the UK 'BIM Revolution'. In *Engineering Project Organization Conference*.

Dall'Asta, E., et al. (2015). Network Design and Quality Checks in Automatic Orientation of Close-Range Photogrammetric Blocks. *Sensors*, pages 7985–8008.

Danion, F., et al. (2003). Stride variability in human gait: the effect of stride frequency and stride length. *Gait & Posture*, 18(1):69–77.

Davison, A. J., et al. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE TPAMI*, 29(6):1052–1067.

De, S., et al. (2016). *Image Segmentation: A Review*, pages 29–40. Springer International Publishing, Cham.

De Boor, C., et al. (1978). *A practical guide to splines*, volume 27. Springer-Verlag New York.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons.

Deb, K. (2011). *Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction*.

Dedieu, J., et al. (1994). Algorithms for ordering unorganized points along parametrized curves. *Numerical Algorithms*, 6:169–200.

Deems, J. S., et al. (2013). Lidar Measurement of Snow Depth: A Review. *Journal of Glaciology*, 59(215):467–479.

Degol, J., et al. (2016). Geometry-Informed Material Recognition. In *IEEE CVPR Proceedings*, pages 1554–1562.

Delicado, P. (1998). Principal Curves and Principal Oriented Points. Technical report.

Dermanis, A. (1994). The photogrammetric inner constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 49(1):25–39.

Dey, T. K. (2006). *Curve and surface reconstruction: algorithms with mathematical analysis*. Cambridge University Press.

Dickscheid, T., et al. (2009). Evaluating the Suitability of Feature Detectors for Automatic Image Orientation Systems. In *International Conference of Computer Vision Systems*, Liege.

Dimitrov, A., et al. (2014). Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections. *Advanced Engineering Informatics*, 28:37–49.

Dimitrov, A., et al. (2015). Segmentation of building point cloud models including detailed architectural/structural features and MEP systems. *Automation in Construction*, 51:32–45.

Dimitrov, A., et al. (2016). Non-Uniform B-Spline Surface Fitting from Unordered 3D Point Clouds for As-Built Modeling. *Computer-Aided Civil and Infrastructure Engineering*, 31(7):483–498.

Dissanayake, M., et al. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241.

Donahue, J., et al. (2014). DeCAF : A Deep Convolutional Activation Feature for Generic Visual Recognition. 32.

Douglas, D. H., et al. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122.

Durrant-Whyte, H., et al. (2006). Simultaneous localization and mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–108.

El-Hakim, S. F. (2004). Photogrammetric Applications. In McGlone, C. J., et al., editors, *Manual of Photogrammetry*, chapter 14. 5th edition.

Engels, C., et al. (2006). Bundle adjustment rules. In *ISPRS Photogrammetric Computer Vision*.

Eyre, M., et al. (2016). Evaluation of Automated Underground Mapping Solutions for Mining and Civil Engineering Applications. *Journal of Applied Remote Sensing*, 10(4).

Falenty, A., et al. (2014). Formation and properties of ice XVI obtained by emptying a type sII clathrate hydrate. *Nature*, 516(7530):231–233.

Fan, B., et al. (2012). Rotationally invariant descriptors using intensity order pooling. *IEEE TPAMI*, 34(10):2031–2045.

Faraday, M. (1859). Note on regelation. *Proceedings of the Royal Society of London*, 10:440–450.

Fathi, H., et al. (2015). Automated as-built 3D reconstruction of civil infrastructure using computer vision: Achievements, opportunities, and challenges. *Advanced Engineering Informatics*, 29(2):149–161.

FIBT (2013). Skeleton Rules, Fédération Internationale de Bobsleigh et de Tobogganing.

Filip, D. J., et al. (1989). Procedurally representing lofted surfaces. *IEEE computer graphics and applications*, 9(6):27–33.

Fischer, A., et al. (2013). The symmetric quadratic traveling salesman problem. *Mathematical Programming*, 1(142):205–254.

Fischler, M. A., et al. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Fitzgibbon, A. W., et al. (1998). Automatic camera recovery for closed or open image sequences. In *ECCV*. Springer, Berlin-Heidelberg.

Fletcher, N. H. (1970). *The Chemical Physics of Ice*. Cambridge Monographs on Physics. Cambridge University Press.

Flynn, P. J., et al. (1989). On reliable curvature estimation. In *IEEE CVPR*, pages 110–116.

Focus, F. (2017). https://www.faro.com/en-gb/products/construction-bim-cim/faro-focus/ .

Fonstad, M. A., et al. (2013). Topographic structure from motion : a new development in photogrammetric measurement. *Earth Surface Processes and Landforms*, 38:421–430.

Forlani, G., et al. (2015). Where is photogrammetry heading to ? State of the art and trends. *Rendiconti Lincei*, 26(1):85–96.

Förstner, W. (1993). A future of photogrammetric research. *NGT Geodesia*, 93(8):372–383.

Förstner, W. (2002). Computer Vision and Photogrammetry - Mutual Questions: Geometry, Statistics and Cognition. In *International Symposium Photogrammetry meets Geoinformatics*.

Förstner, W., et al. (2004). On the completeness of coding with image features. *Computing*, 22:761–767.

Förstner, W., et al. (1987). A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, pages 281–305.

Förstner, W., et al. (2016). *Photogrammetric Computer Vision*. Springer.

Fraser, C. S. (1982). Optimization of precision in close-range photogrammetry. *Photogrammetric Engineering and Remote Sensing*.

Fraser, C. S. (1983). Network Design Optimisation in non-Topographic Photogrammetry. *International Archives of Photogrammetry and Remote Sensing*, 25:296–307.

Fraser, C. S. (1984). Network Design Considerations for non-Topographic Photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 50(8):1115–1126.

Fraser, C. S. (1996). Network Design. In Atkinson, K. B., editor, *Close Range Photogrammetry and Machine Vision*, pages 256–281.

Fraser, C. S. (2013). Automatic Camera Calibration in Close Range Photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 79(4):381–388.

Fraundorfer, F., et al. (2004). Evaluation of local detectors on non-planar scenes. In *28th workshop of AAPR*, pages 125–132.

Fraundorfer, F., et al. (2005). A novel performance evaluation method of local detectors on non-planar scenes. In *IEEE CVPR*, page 33.

Fuentes-Pacheco, J., et al. (2015). Visual simultaneous localization and mapping : a survey. *Artificial Intelligence Review*, 43:55–81.

Furukawa, Y., et al. (2010). Accurate, Dense, and Robust Multiview Stereopsis. *IEEE TPAMI*, 32(8):1362–1376.

Garach, L., et al. (2014). Mathematical formulation and preliminary testing of a spline approximation algorithm for the extraction of road alignments. *Automation in Construction*, 47:1–9.

Garrett, M., et al. (2010). Computer-Generated Three-Dimensional Training Environments: The Simulation , User , and Problem-Based Learning (SUPL) Approach. *International Journal of Gaming and Computer-Mediated Simulations*.

Geomagic Studio (2012). *3D Systems, Inc*. https://www.3dsystems.com/.

GeoSLAM (2015). Zeb1. *https://geoslam.com/hardware/zeb-1/*.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Psychology Press.

Gikas, V., et al. (2008). Determining Rail Track Axis Geometry Using Satellite and Terrestrial Geodetic Data. *Survey Review*, 40(310):392–405.

Gikas, V., et al. (2012). A Novel Geodetic Engineering Method for Accurate and Automated Road/Railway Centerline Geometry Extraction Based on the Bearing Diagram and Fractal Behavior. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):115–126.

GitHub DBAT Issue (2016). Mark Points Uncertainty: converting from PhotoScan to PhotoModeler to DBAT structure. *https://github.com/niclasborlin/dbat/issues/1*.

Godin, G., et al. (2001). An Assessment of Laser Range Measurement on Marble Surfaces. In *Conferenfce on Optical Measurement Techniques*, Austria.

Gong, C., et al. (2016). Analysis of Performance Indices for Simulated Skeleton Descents. In *11th conference of the International Sports Engineering Association, ISEA*, volume 147, pages 712–717.

Goshtasby, A. (1986). Piecewise linear mapping functions for image registration. *Pattern Recognition*, 19(6):459–466.

Goshtasby, A. (1988). Image registration by local approximation methods. *Image and Vision Computing*, 6(4):255–261.

Grafarend, E. W. (1974). Optimization of Geodetic Networks. *Bolletino di Geodesia a Science Affini*, 33(4):351–406.

Granshaw, S. I. (1980). Bundle Adjustment Methods in Engineering Photogrammetry. *The Photogrammetric Record*, 10(56):181–207.

Granshaw, S. I. (2016a). Photogrammetric Terminology: Third Edition. *The Photogrammetric Record*, 31(154):210–252.

Granshaw, S. I. (2016b). Terminological Dichotomies. *The Photogrammetric Record*, 31(June):116–120.

Granshaw, S. I. (2018). Structure from motion: origins and originality. *The Photogrammetric Record*, 33(161):6–10.

Granshaw, S. I., et al. (2015). Editorial: Computer Vision and Photogrammetry: Interaction or Introspection? *The Photogrammetric Record*, 30(149):3–7.

Grimm, A. (2007). The origin of the term photogrammetry. In *Photogrammetric Week*, pages 53–60.

Gruen, A. (1996). Least squares matching: a fundamental measurement algorithm. *Close Range Photogrammetry and Machine Vision*, pages 217–255.

Gruen, A. (2012). Development and Status of Image Matching in Photogrammetry. *Photogrammetric Record*, 26(137):36–57.

Gruen, A., et al. (2001). *Calibration and Orientation of Cameras in Computer Vision*.

Guidi, G., et al. (2009). Range sensors on marble surfaces: quantitative evaluation of artifacts. *Videometrics, Range Imaging, and Applications Xi*.

Hale, K. S., et al. (2014). *Handbook of virtual environments: Design, implementation, and applications*. CRC Press.

Hane, C., et al. (2013). Joint 3D scene reconstruction and class segmentation. In *IEEE CVPR*, pages 97–104.

Harary, G., et al. (2012). 3D Euler Spirals for 3D Curve Completion. *Computational Geometry*, 45(3):115–126.

Harary, G., et al. (2014). Context-based coherent surface completion. *ACM Transactions on Graphics (TOG)*, 33(1):5.

Harris, C., et al. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 147–151. Manchester, UK.

Hartley, R. I., et al. (1993). Relationship between photogrammmetry and computer vision. In *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*.

Hartmann, W., et al. (2016). Recent developments in large-scale tie-point matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115(May):1–27.

Hastedt, H. (2004). Monte-Carlo-Simulation in Close-Range Photogrammetry. In *ISPRS Symposium Commission V, Istanbul, Turkey, ISPRS Archives*, volume 35, pages 18–23.

Hastie, T., et al. (1989). Principal Curves. *Journal of the American Statistical Association*, 84(406):502–516.

Havemann, S., et al. (2013). Curvature-controlled curve editing using piecewise clothoid curves. *Computers and Graphics*, 37(6):764–773.

Heald, M. A. (1985). Rational approximations for the Fresnel integrals. *Mathematics of Computation*, 44(170):459–461.

Heikkinen, J. (2005). *The Circular Imaging Block in Close Range Photogrammetry*. PhD thesis, Helsinki University of Technology.

Heinly, J., et al. (2012). Comparative Evaluation of Binary Features. In *ECCV*, pages 759–773. Springer.

Held, M. (2001). Vroni: An engineering approach to the reliable and efficient computation of voronoi diagrams of points and line segments. *Computational Geometry*, 18(2):95 – 123.

Hesch, J. a., et al. (2013). Camera-IMU-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201.

Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR Conference*, pages 807–814, San Diego, CA.

Hirschmuller, H. (2008). Stereo processing by Semiglobal Matching and Mutual Information. *IEEE TPAMI*, 30(2):328–341.

Hobbs, P. V. (2010). *Ice physics*. Oxford university press.

Hokuyo (2015). UTM-30LX. *https://www.hokuyo-aut.jp/search/single.php?serial=169*.

Holgado-Barco, A., et al. (2014). Semiautomatic Extraction of Road Horizontal Alignment from a Mobile LiDAR System. *Computer-Aided Civil and Infrastructure Engineering*, 30:1–12.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*.

Hoppe, C., et al. (2012). Online Feedback for Structure-from-Motion Image Acquisition. In *British Machine Vision Conference*, pages 1–12.

Horn, B. K. P. (1988). Motion fields are hardly ever ambiguous. *International Journal of Computer Vision*, 1(3):259–274.

Hoschek, J. (1988). Intrinsic parametrization for approximation. *Computer Aided Geometric Design*, 5(1):27–31.

Hosseininaveh Ahmadabadian, A., et al. (2014). Stereo-imaging network design for precise and dense 3d reconstruction. *The Photogrammetric Record*, 29(147):317–336.

Hosseininaveh Ahmadabadian, A., et al. (2013). A comparison of dense matching algorithms for scaled surface reconstruction using stereo camera rigs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 78:157–167.

Hough, P. (1962). Method and means for recognizing complex patterns. *US Patent 3,069,654*.

Hubbard, M. (2013). Luge track safety. *Sports Engineering*, 16(2):123–135.

Hubbard, M., et al. (1989). Three-Dimensional Bobsled Turning Dynamics. *Journal of Applied Biomechanics*, 5:222–237.

Huffman, R. K., et al. (1996). A motion based virtual reality training simulator for bobsled drivers. In *Engineering of Sport*, pages 195–203, Rotterdam.

IBG+PARTNER (2017). Engineering Company, Leipzig, Germany. *http://ibg-partner.com [Accessed on 01/09/2017]*.

IBSF (2016). International Skeleton Rules 2016.

IBSF (2017). Internation Bobsleigh & Skeleton Federation. *http://www.ibsf.org/ [Accessed on 01/09/2017]*.

Ihrke, I., et al. (2010). Transparent and Specular Object Reconstruction. *Computer Graphics Forum*, 29(8):2400–2426.

Imran, M., et al. (2006). GPS – GIS-Based Procedure for Tracking Vehicle Path on Horizontal Alignments. *Computer-Aided Civil and Infrastructure Engineering*, 21:383–394.

Jaakkola, A., et al. (2014). Measurement of Snow Depth Using a Low-Cost Mobile Laser Scanner. *IEEE Geoscience and Remote Sensing Letters*, 11(3):587–591.

Jaud, M., et al. (2016). Assessing the Accuracy of High Resolution Digital Surface Models Computed by PhotoScan® and MicMac® in Sub-Optimal Survey Conditions. *Remote Sensing*, 8(6).

Javernick, L., et al. (2014). Geomorphology Modeling the topography of shallow braided rivers using Structure-from-Motion photogrammetry. *Geomorphology*, 213:166–182.

Jha, M. K., et al. (2006). *Intelligent road design*, volume 19. WIT Press.

Jiménez, F., et al. (2009). Measurement uncertainty determination and curve-fitting algorithms for development of accurate digital maps for advanced driver assistance systems. *Transportation Research PartC*, 17(3):225–239.

Jong, J.-C., et al. (1999). Cost functions for optimizing highway alignments. *Transportation Research Record: Journal of the Transportation Research Board*, (1659):58–67.

Jong, J.-C., et al. (2003). An evolutionary model for simultaneously optimizing three-dimensional highway alignments. *Transportation Research Part B: Methodological*, 37(2):107–128.

Juan, L., et al. (2009). A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152.

Jung, J., et al. (2014). Automation in Construction Productive modeling for development of as-built BIM of existing indoor structures. *Automation in Construction*, 42:68–77.

Jung, S. H., et al. (2001). Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *IEEE CVPR Proceedings*, volume 2, pages II–732–II–737 vol.2.

Kabanikhin, S. I. (2008). Definitions and examples of inverse and ill-posed problems. *Journal of Inverse and Ill-posed Problems*, 16(4):317.

Kahl, F. (1999). Critical Motions and Ambiguous Euclidean Reconstructions in Auto-Calibration. *7th ICCV*, pages 469–475.

Kahl, F., et al. (2002). Critical curves and surfaces for euclidean reconstruction. *ECCV 2002*, (1):447–462.

Kahl, F., et al. (2000). Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision*, 13(iii):131–146.

Kanatani, K. (2004). Uncertainty Modeling and Model Selection for Geometric Inference. *IEEE TPAMI*, 26(10):1307–1319.

Kane, T. R., et al. (1985). *Dynamics, theory and applications*. McGraw Hill.

Kang, M.-W., et al. (2012). Applicability of highway alignment optimization models. *Transportation Research Part C: Emerging Technologies*, 21(1):257–286.

Kang, Z., et al. (2014). Continuous Extraction of Subway Tunnel Cross Sections Based on Terrestrial Point Clouds. *Remote Sensing*, 6(1):857–879.

Karamanou, A., et al. (2009). Precise Estimation of Road Horizontal and Vertical Geometric Features Using Mobile Mapping Techniques. *Boletim de Ciencias Geodesicas*, 15(5):762–775.

Kazhdan, M., et al. (2006). Poisson Surface Reconstruction. In *Eurographics Symposium on Geometry Processing*.

Kazhdan, M., et al. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29.

Kegl, B., et al. (2000). Learning and Design of Principal Curves. *IEEE TPAMI*, 8(3):281–297.

Kelly, A., et al. (2000). Design and construction of a bobsled driver training simulator. *Sports Engineering*, 3:13–24.

Kessler, D. A. (2001). Surface physics: A new crack at friction. *Nature*, 413(6853):260–261.

Kietzig, A.-m., et al. (2010). Physics of ice friction. *Journal of Applied Physics*, 107.

Kimia, B., et al. (2003). Euler spiral for shape completion. *International Journal of Computer Vision*, 54:159–182.

Klette, R. (2014). *Concise Computer Vision*. Springer-Verlag, London.

Koegel, M., et al. (2011). A comparison of vehicular trajectory encoding techniques. *2011 the 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net'2011*, pages 87–94.

Koenderink, J. J. (1990). *Solid Shape*. MIT Press, Cambridge, MA, USA.

Konolige, K., et al. (2011). Large Scale Visual Odometry for Rough Terrain. In Kaneko, M., et al., editors, *Springer (STAR)*, pages 201–212. Springer.

Kraus, K. (1997). *Photogrammetry*, volume 2. Ferdinand Dummlers Verlag.

Kraus, K. (2007). *Photogrammetry: geometry from images and laser scans*. Walter de Gruyter, second edition.

Kuhn, W. (2013). *Fundamentals of Road Design*.

Ladicky, L., et al. (2012). Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, pages 1–12.

Leach, R. (2011). *Optical measurement of surface topography*, volume 14. Springer.

Leach, R., editor (2013). *Characterisation of Areal Surface Texture*. Springer.

Lee, Y., et al. (2009). Optimization method for highway horizontal alignment design. *Journal of Transportation Engineering*, 135(4):217–224.

Leung, T., et al. (2001). Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. *International Journal of Computer Vision*, 43(1):29–44.

Leutenegger, S., et al. (2011). BRISK : Binary Robust Invariant Scalable Keypoints. In *IEEE ICCV*, pages 2548–2555.

Levien, R. (2008). The Euler spiral: a mathematical history. *UCB/EECS-2008-111*.

Levy, R., et al. (2007). Virtual Reality Simulation: Bobsled and Luge. In *IACSS International Symposium Computer Science in Sport*.

Lewiner, T., et al. (2005). Curvature and torsion estimators based on parametric curve fitting. *Computers & Graphics*, 29:641–655.

Lewis, J. P. (1995). Fast normalized cross-correlation. In *Vision interface*, volume 10, pages 120–123.

Li, R., et al. (1997). Quantitative photogrammetric analysis of digital underwater video imagery. *IEEE Journal of Oceanic Engineering*, 22(2):364–375.

Li, Y., et al. (2004). Lazy Snapping. *ACM Transactions on Graphics*, 23(3):303–308.

Li, Z., et al. (2012). Automated Identification and Extraction of Horizontal Curve Information from Geographic Information System Roadway Maps. *Transportation Research Record: Journal of the Transportation Research Board*, 2291:80–92.

Lin, W.-Y., et al. (2010). Robust and Accurate Curvature Estimation Using Adaptive Line Integrals. *EURASIP Journal on Advances in Signal Processing*, 2010.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116.

Liu, H., et al. (2008). A Unified Curvature Definition for Regular, Polygonal, and Digital Planar Curves. *International Journal of Computer Vision*, 80(1):104–124.

Liu, J.-W., et al. (2010). Videogrammetric system for dynamic deformation measurement during metal sheet welding processes. *Optical Engineering*, 49(3).

Liu, Y., et al. (2005). Reconstructing B-spline curves from point clouds - A tangential flow approach using least squares minimization. *Proceedings - International Conference on Shape Modeling and Applications, SMI'05*, 2005:4–12.

Lobo, J., et al. (2007). Relative Pose Calibration Between Visual and Inertial Sensors. *The International Journal of Robotics Research*, 26(6):561–575.

Lombard, M., et al. (1997). At the Heart of It All : The Concept of Presence. *Journal of Computer-Mediated Communication*, pages 1–26.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Luhmann, T. (2011). 3D Imaging - How to Achieve Highest Accuracy. *Videometrics, Range Imaging, and Applications Xi*, 8085.

Luhmann, T., et al. (2016). Sensor modelling and camera calibration for close-range photogrammetry. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:37–46.

Luhmann, T., et al. (2014). *Close-Range Photogrammetry and 3D Imaging*.

Luhmann, T., et al. (2006). *Close Range Photogrammetry: Principles, Methods and Applications*.

Luhmann, T., et al. (2013). *Close range photogrammetry: principles, techniques and applications*. Whittles.

Luth, H. (2001). *Surface and Interface Physics: Its Definition and Importance*, pages 1–31. Springer, Berlin, Heidelberg.

Maier, G. (2014). Optimal arc spline approximation. *Computer Aided Geometric Design*, 31(5):211–226.

Mair, E., et al. (2011). Spatio-temporal initialization for IMU to camera registration. *2011 IEEE International Conference on Robotics and Biomimetics*, pages 557–564.

Marhl, M., et al. (1996). Extensions of sweep surface constructions. *Computer & Graphics*, 20(6):893–903.

Marinelli, G., et al. (2017). Mobile mapping systems and spatial data collection strategies assessment in the identification of horizontal alignment of highways. *Transportation Research Part C*, 79:257–273.

Marler, R. T., et al. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395.

Marshall, A. (1989). Network Design and Optimisation in Close Range Photogrammetry. Technical report, The University of new South Wales.

Mason, S. (1994). *Expert System-Based Design of Photogrammetric Networks*. PhD thesis, ETH.

Mason, S. (1995). Expert System-Based Design of Close-Range Photogrammetric Networks. *Isprs Journal of Photogrammetry and Remote Sensing*, 50(5):13–24.

Mason, S. (1997). Heuristic Reasoning Strategy for Automated Sensor Placement. *Photogrammetric Engineering & Remote Sensing*, 63(9):1093–1102.

Mason, S., et al. (1992). On the Representation of Close-Range Network Design Knowledge. *International Archives of Photogrammetry and Remote Sensing*, XXIX(V):446–453.

Matas, J., et al. (2002). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions.

Matlab (2017a). version 9.0.0 (R2017a), Cubic smoothing spline (csaps), https://uk.mathworks.com/help/curvefit/csaps.html. *The MathWorks Inc.*

Matlab (2017b). version 9.0.0 (R2017a), Cubic spline interpolation with end conditions (csape), https://uk.mathworks.com/help/curvefit/csape.html. *The MathWorks Inc.*

Matlab (2017c). version 9.0.0 (R2017a), Find Local Maxima (findpeaks), https://uk.mathworks.com/help/signal/ref/findpeaks.html. *The MathWorks Inc.*

Matlab (2017d). version 9.0.0 (R2017a), Find minimum of constrained nonlinear multivariable function (fmincon), https://uk.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html. *The MathWorks Inc.*

Matlab (2017e). version 9.0.0 (R2017a), Find Pareto front of multiple fitness functions using genetic algorithm (gamultiobj), https://uk.mathworks.com/help/gads/gamultiobj.html. *The MathWorks Inc.*

Matlab (2017f). version 9.0.0 (R2017a), Least-squares spline approximation (spap2), https://uk.mathworks.com/help/curvefit/spap2.html. *The MathWorks Inc.*

Matlab (2017g). version 9.0.0 (R2017a), Smooth response data (smooth), https://uk.mathworks.com/help/curvefit/smooth.html. *The MathWorks Inc.*

MATLAB (2017). *version 9.2.0 (R2017a)*. The MathWorks Inc.

McCarthy, J., et al. (2006). A proposal for the dartmouth summer research project on artificial intelligence, August 31, 1955. *AI magazine*, 27(4):12.

McCrae, J., et al. (2009). Sketching piecewise clothoid curves. *Computers & Graphics*, 33(4):452–461.

McGlone, C. J., editor (2004). *Manual of Photogrammetry*. ASPRS, fifth edition.

McGlone, C. J., editor (2013). *Manual of Photogrammetry*. ASPRS, sixth edition.

McLaren, R. (2016). WADA Investigation of Sochi Allegations. Technical report.

Meek, D. (1991). A guided clothoid spline. *Computer Aided Geometric Design*, 8:163.–174.

Meek, D., et al. (1989). The use of Cornu spirals in drawing planar curves of controlled curvature. *Journal of Computational and Applied Mathematics*, 25:69–78.

Meek, D. S., et al. (1992). Clothoid spline transition spirals. *Mathematics of Computation*, 59(199):117–117.

Meek, D. S., et al. (1995). Approximating smooth planar curves by arc splines. *Journal of Computational and Applied Mathematics*, 59(2):221–231.

Micheletti, N., et al. (2015). Structure from motion (SFM) photogrammetry. In Clarke, L. E., et al., editors, *Geomorphological Techniques*, chapter 2.

Microsoft Kinect (2010). https://developer.microsoft.com/en-us/windows/kinect.

Microstrain (2015). 3DM-GX2. *http://www.microstrain.com/inertial/3dm-gx2*.

Mikhail, E. M., et al. (1976). Observations and Least Squares.

Mikhail, E. M., et al. (2001). *Introduction to Modern Photogrammetry*.

Mikolajczyk, K., et al. (2004). Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86.

Mikolajczyk, K., et al. (2005). A performance evaluation of local descriptors. *IEEE TPAMI*, 27(10):1615–1630.

Miksik, O., et al. (2012). Evaluation of Local Detectors and Descriptors for Fast Feature Matching. In *IEEE ICPR12*.

Miles, H. C., et al. (2012). A review of virtual environments for training in ball sports. *Computers and Graphics*, 36(6):714–726.

Mittal, A., et al. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708.

Mittal, A., et al. (2013). Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212.

Mokhtarian, F., et al. (1992). A theory of multiscale, curvature-based shape representation for planar curves. *IEEE TPAMI*, 14(8):789–805.

Moons, T., et al. (2008). 3D Reconstruction from Multiple Images Part 1 : Principles. *Foundations and Trends in Computer Graphics and Vision*, 4(4):287–398.

Moreels, P., et al. (2007). Evaluation of Features Detectors and Descriptors based on 3D Objects. *International Journal of Computer Vision*, 73(3):263–284.

Morris, D. D., et al. (2002). Feature Localization Error in 3D Computer Vision. In Winkler, J., et al., editors, *Uncertainty in Geometric Computations*, chapter Chapter 9, pages 107–117. Springer US, Boston, MA.

Mossner, M., et al. (2011). An approximate simulation model for initial luge track design. *Journal of Biomechanics*, 44(5):892–896.

Mostegel, C., et al. (2016). UAV-based autonomous image acquisition with multi-view stereo quality assurance by confidence prediction. In *IEEE CVPR Workshops Proceedings*, pages 1–10.

MXROAD (2018). *https://www.bentley.com/en/products/product-line/civil-design-software/mxroad*. Bentley Systems, Incorporated, Exton, PA 19341, United States.

Myronenko, A., et al. (2010). Point set registration: Coherent point drift. *IEEE TPAMI*, 32(12):2262–2275.

Nayar, S. K., et al. (1991). Surface Reflection: Physical and Geometrical Perspectives. *IEEE TPAMI*, 13(7):611–634.

Neto, J. A., et al. (2002). Using Colors to Detect Structural Components in Digital Pictures. *Computer Aided Civil and Infrastructure Engineering*, 17:61–67.

Newcombe, R. A. (2012). *Dense Visual SLAM*. PhD thesis, Imperial College London.

Nguyen, H. H., et al. (2014). Realistic road path reconstruction from GIS data. *Computer Graphics Forum*, 33(7).

Nguyen, T. A., et al. (2008). VICUR : A human-vision-based algorithm for curve reconstruction. *Robotics and Computer-Integrated Manufacturing*, 24:824–834.

Nister, D., et al. (2004). Visual odometry. In *IEEE Computer Society CVPR Conference*, pages 652–659.

Northam, G. (2000). Simulation Fidelity – Getting in Touch with Reality What is Fidelity ? In *Proceedings of SimTecT*.

Norwegian Mapping Authority ©Kartverket (2017). https://kartverket.no. *http://www.norgeskart.no*.

Novak, D. (2014). *Photogrammetric Generation and Registration of Point Clouds*. PhD thesis.

Nützi, G., et al. (2011). Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM. *Journal of Intelligent & Robotic Systems& Robotic Systems*, 61(1):287–299.

Oculus Rift (2016). https://www.oculus.com/.

Olague, G., et al. (2007). Development of a Practical Photogrammetric Network Design Using Evolutionary Computing. *The Photogrammetric Record*, 22(117):22–38.

Oliensis, J. (2001). A Critique of Structure-from-Motion Algorithms. *Computer Vision and Image Understanding*, 84(3):407–408.

Olympia (2018). *Olympia Sport- und Veranstaltungszentrum Innsbruck GmbH*. https://www.olympiaworld.at/arenen-angebote/die-arenen/olympia-eiskanal/.

OMEGA (2014). *OMEGA MEASUREMENT UNIT*. https://www.omegawatches.com/planet-omega/sport/our-sports/bobsleigh-and-monobob/, and https://www.omegawatches.com/news/news-detail/2314/.

Opplysningen (2017). 1881. *https://kart.1881.no/*.

O'Rourke, J., et al. (1987). Connect-the-dots: A new heuristic. *Computer Vision, Graphics, and Image Processing*, 39(2):258–266.

Panasonic Lumix DMC-G1 (2017). *http://www.panasonic.com/au/support/manual-lumix-g-cameras.html*.

Papert, S. (1966). The summer vision project. Technical report, MIT.

Parian, J. A. (2007). *Sensor Modeling, Calibration and Point Positioning with Terrestrial Panoramic Cameras*. PhD thesis.

Patraucean, V., et al. (2015). State of research in automatic as-built modelling. *Advanced Engineering Informatics*, 29(2):162–171.

Pavlis, N. K., et al. (2012). The development and evaluation of the Earth Gravitational Model 2008 (EGM2008). *Journal of Geophysical Research: Solid Earth*, 117(B4).

Pech-Pacheco, J. L., et al. (2000). Diatom autofocusing in brightfield microscopy: a comparative study. In *IEEE ICCV*, volume 3, pages 314–317.

Pellegrino, O. (2009). An analysis of the effect of roadway design on driver's workload. *The Baltic journal of road and bridge engineering*, 4(2):45–53.

Pertuz, S., et al. (2013). Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415–1432.

Petrenko, V., et al. (1999). *Physics of Ice*.

PhotoModeler Scanner (2016). Eos Systems Inc. *http://www.photomodeler.com/products/scanner/defau*.

PhotoScan (2018). Agisoft LLC. *www.agisoft.com*.

Photosynth (2008). Microsoft. *https://blogs.msdn.microsoft.com/photosynth/2017/0*.

Piegl, L. A., et al. (1996). Algorithm for approximate NURBS skinning. *Computer Aided Design*, 28(9):699–706.

Piegl, L. A., et al. (1997). *The NURBS book*. Springer, second edition.

Piegl, L. A., et al. (2002). Surface skinning revisited. *The Visual Computer*, 18:273–283.

Pierrot-Deseilligny, M., et al. (2011). APERO, an Open Source Bundle Adjustment for Automatic Calibration and Orientations of Sets of Images.

Pierrot-Deseilligny, M., et al. (2006). A multiresolution and optimization-based image matching approach: an application to surface reconstruction from SPOT5-HRS stereo imagery. *ISPRS Archives*.

Pix4D (2017). Professional drone mapping software, https://pix4d.com/.

Poirier, L., et al. (2011a). Experimental analysis of ice friction in the sport of bobsleigh. *Sports Engineering*, 14:67–72.

Poirier, L., et al. (2011b). Ice Hardness in Winter Sports. *Cold Regions Science and Technology*, 67(3):129–134.

Poirier, L., et al. (2011c). Getting a Grip on Ice Friction. In *21st International Offshore and Polar Engineering Conference*, volume 8, pages 1071–1077.

Pollefeys, M., et al. (2004). Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232.

Pollefeys, M., et al. (2008). Detailed Real-Time Urban 3D Reconstruction From Video. *International Journal of Computer Vision*, 78(2-3):143–167.

Polya, G. (1945). *How to Solve It: A New Aspect of Mathematical Model*. Princeton University Press.

Pope, A., et al. (2014). Using in situ spectra to explore Landsat classification of glacier surfaces. *International Journal of Applied Earth Observation and Geoinformation*, 27:42–52.

Popov, V. L. (2016). What does friction really depend on? Robust governing parameters in contact mechanics and friction. *Physical Mesomechanics*, 19(2):115–122.

Pottmann, H., et al. (2002). Approximation with active B-spline curves and surfaces. In *10th Pacific Conference on Computer Graphics and Applications*, pages 8–25.

Powell, M. J. (1978). *A fast algorithm for nonlinearly constrained optimization calculations*. Springer.

Prince, S. J. D. (2012). *Computer vision : models , learning and inference*.

Prokop, A. (2008). Assessing the applicability of terrestrial laser scanning for spatial snow depth measurements. *Cold Regions Science and Technology*, 54(3):155–163.

Pu, S., et al. (2009). ISPRS Journal of Photogrammetry and Remote Sensing Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6):575–584.

Pyeongchang (2018). *https://www.olympic.org/pyeongchang-2018*.

Python (1991). *version 3.3*. Python Software Foundation, http://www.python.org.

Quan, L. (2010). *Image-Based Modeling*. Springer Science-Business-Media.

Rabbani, T., et al. (2006). Segmentation of point clouds using smoothness constraint. *ISPRS Archives - Commission V Symposium 'Image Engineering and Vision Metrology'*, 36(5):248–253.

RAENG (2018). *Athlete or Machine?* The Royal Academy of Engineering, https://www.raeng.org.uk/publications/other/bob-pack-web.

Rao, S. S. (2009). *Engineering Optimization: Theory and Practice: Fourth Edition*. John Wiley and Sons, fourth edition.

Rashidi, A., et al. (2013). Optimized selection of key frames for monocular videogrammetric surveying of civil infrastructure. *Advanced Engineering Informatics*, 27(2):270–282.

Rashidi, A., et al. (2015). An analogy between various machine-learning techniques for detecting construction materials in digital images. *KSCE Journal of Civil Engineering*, 00(0000):1–11.

RealityCapture (2016). Capturingreality s.r.o. *www.capturingreality.com*.

Remondino, F., et al. (2006). Image-based 3D modelling: A review. *The Photogrammetric Record*, 21(115):269–291.

Remondino, F., et al. (2012). Low-Cost and Open-Source Solutions for Automated Image Orientation – A Critical Overview. In *4th International Conference, EuroMed 2012*, pages 40–54.

Remondino, F., et al. (2014). State of the Art in High Density Image Matching. *The Photogrammetric Record*, 29(146):144–166.

Rempfler, G. S. (2015). *Entwicklung eines Bobsimulators*. PhD thesis, ETH.

Rempfler, G. S., et al. (2016). A bobsleigh simulator software. *Multibody System Dynamics*, 36:257–278.

Ren, X., et al. (2003). Learning a classification model for segmentation. In *IEEE ICCV*, page 10.

Rey-otero, I., et al. (2015). Is Repeatability an Unbiased Criterion for Ranking Feature Detectors ? *SIAM Journal on Imaging Sciences*, 8(4):2558–2580.

Rhino 5 (2018). *3D CAD software*. Robert McNell & Associates, https://www.rhino3d.com.

RICS (2014). What is BIM?

Roche, J., et al. (2008). *An Analysis of the Interaction Between Slider Physique and Descent Time for the Bob Skeleton (P153)*, pages 101–109. Springer Paris, Paris.

Romero-Zaliz, R., et al. (2015). Minimizing B-spline knots in representative road axis from GPS points cloud. *Mathematical Methods in the Applied Sciences*, 39(16):4773–4779.

Rosenberg, R. (2005). Why Is Ice Slippery? *Physics Today*, 58(12):50.

Rother, C., et al. (2004). " GrabCut " — Interactive Foreground Extraction using Iterated Graph Cuts. 1(212):309–314.

Rothermel, M., et al. (2012). Sure: Photogrammetric surface reconstruction from imagery. In *Proceedings LC3D Workshop, Berlin*, volume 8.

Rublee, E., et al. (2011). ORB: An efficient alternative to SIFT or SURF. In *IEEE ICCV*, pages 2564–2571.

Rumpler, M., et al. (2017). Evaluations on multi-scale camera networks for precise and geo-accurate reconstructions from aerial and terrestrial images with user guidance. *Computer Vision and Image Understanding*, 157:255–273.

Russell, S. J., et al. (1995). *Artificial Intelligence A Modern Approach*.

Saadatseresht, M., et al. (2005). Automatic Camera Placement in Vision Metrology Based On A Fuzzy Inference System. *Photogrammetric Engineering & Remote Sensing*, 71(12):1375–1385.

Sabbioni, E., et al. (2016). Bobsleigh and Skeleton. In *The Engineering Approach to Winter Sports*, chapter 7, pages 183–276.

SAIT (2012). *Whistler Sliding Centre Sled Trajectory & Track Construction Study*. Southern Alberta Institute of Technology.

Sareen, K. K., et al. (2011). Hierarchical data clustering approach for segmenting colored three-dimensional point clouds of building interiors. *Optical Engineering*, 50(7):77003.

Scaramuzza, D., et al. (2011). Visual Odometry Part I: The First 30 Years and Fundamentals. *IEEE Robotics & Automation Magazine*, 18(4):80–92.

Scharstein, D., et al. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42.

Schlogelhofer, H. (1989). *Qualitats- und Wirtschaftlichkeitsmodelle fur die Ingenieurphotogrammetrie*. PhD thesis.

Schmid, C., et al. (1998). Comparing and Evaluating Interest Points. In *Sixth IEEE ICCV*, pages 230–235.

Schmid, H. (1958). Eine allgemeine analytische Losung fur die Aufgabe der Photogrammetrie. *Bildmessung und Luftbildwesen*, 26(27):4.

Schmidt, A., et al. (2015). The Impact of the Image Feature Detector and Descriptor Choice on Visual SLAM Accuracy. *Image Processing & Communications Challenges*, pages 203–210.

Schnabel, R., et al. (2007). Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226.

Scott, W. R., et al. (2003). View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys*, 35(1):64–96.

Seitz, S. M., et al. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE CVPR*, pages 519–526.

Seo, Y.-H., et al. (2008). Optimal keyframe selection algorithm for three-dimensional reconstruction in uncalibrated multiple images. *Optical Engineering*, 47(5):53201.

Sharif Razavian, A., et al. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE CVPR conference workshops*, pages 806–813.

Sherman, W. R., et al. (2002). *Understanding virtual reality: Interface, application, and design*. Elsevier.

Shi, J., et al. (2000). Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905.

Shum, H.-Y., et al. (2008). *Image-based rendering*. Springer Science-Business-Media.

Sieberth, T., et al. (2015). UAV Image Blur-Its Influence and Ways to Correct it. *ISPRS Archives*, 40(1):33.

Slocum, R. K., et al. (2017). Simulated Imagery Rendering Workflow for UAS-Based Photogrammetric 3D Reconstruction Accuracy Assessments. *Remote Sensing*, 9(396):1–19.

Smith, A. R. (1978). Color gamut transform pairs. *ACM Siggraph Computer Graphics*, 12(3):12–19.

Snavely, N., et al. (2007). Modeling the World from Internet Photo Collections. *International Journal of Computer Vision*, 80(2):189–210.

Son, H., et al. (2015). As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey. *Advanced Engineering Informatics*, 29(2):172–183.

Son, H., et al. (2014). Classification of major construction materials in construction environments using ensemble classifiers. *Advanced Engineering Informatics*, 28(1):1–10.

SONY DCR-SR300 (2017). *https://esupport.sony.com/US/p/model-home.pl?mdl=DCRSR300*.

Steuer, J. (1992). Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, 42(4):73–93.

Stoer, J. (1982). Curve Fitting with Clothoidal Splines. *Journal of Research of the National Bureau of Standards*, 87(4).

Stout, K. J., et al. (1993). *The Development of Methods for the Characterisation of Roughness in Three Dimensions*. Commission of the European Communities.

Stoye, A. (1990). *Über die räumliche Trassierung von Rennschlitten- und Bobbahnen und ihre Querschnittsprofilierung*. PhD thesis.

Strelow, D., et al. (2004). Motion estimation from image and inertial measurements. *The International Journal of Robotics Research*, 23(12):1157–1195.

Sturm, P. (1997). Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *IEEE CVPR*, pages 1100–1105, San Juan.

Sturm, P. (2000). A case against Kruppa's equations for camera self-calibration. *IEEE TPAMI*, 22(10):1199–1204.

Sturm, P. (2002). Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length. *Image and Vision Computing*, 20(5-6):415–426.

Stutz, D., et al. (2017). Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*.

Sugaya, Y., et al. (2003). Outlier removal for motion tracking by subspace separation. *IEICE Transactions on Information and Systems*, 86(6):1095–1102.

Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*. Springer Science-Business-Media.

Szeliski, R., et al. (1994). Recovering 3D Shape and Motion from Image Streams Using Nonlinear Least Squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28.

Tagliasacchi, A., et al. (2009). Curve Skeleton Extraction from Incomplete Point Cloud. *ACM Transactions on Graphics Graphics*, 28(3).

Tam, G. K., et al. (2013). Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217.

Tang, P., et al. (2010). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, 19(7):829–843.

Tarabanis, K. a., et al. (1995). Survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation*, 11(1):86–104.

Taubin, G. (1991). Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE TPAMI*, 13(11):1115–1138.

Texture Library (2013). *https://www.texturelib.com/*. Dmitriy Chugal [Accessed: 8/2017].

Texturify (2014). *https://www.texturify.com/*. Agata Galewska [Accessed: 8/2017].

The Swatch Group Ltd (2014). Timekeeping technologies at the Sochi 2014 Olympic Winter Games, http://www.swatchgroup.com/ en/services/archive/2014/ timekeeping-technologies-at-the-sochi-2014-olympic-winter-games-1-2.

Thomson, C. (2016). *From Point Cloud to Building Information Model*. PhD thesis, UCL.

Thomson, C., et al. (2015). Automatic Geometry Generation from Point Clouds for BIM. *Remote Sensing*, 7:11753–11775.

Tokuyama, Y. (2000). Skinning-surface generation based on spine-curve control. *The Visual Computer*, 16:134–140.

Tomaszewska, A., et al. (2017). Assessment of the High-Speed Ice Friction of Bob Skeleton Runners. *ICTST 2017 : 19th International Conference on Tribology and Surface Technology*.

Tong, X., et al. (2010). Estimating geometric parameters of highways and railways using least-squares adjustment. *Survey Review*, 42(318):359–374.

Torlegård, K. (1967). *On the determination of interior orientation of close-up cameras under operational conditions using three-dimensional test objects*. PhD thesis, KTH Stockholm.

Torr, P., et al. (1998). Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *IEEE ICCV*, pages 485–491.

Torr, P., et al. (2000). MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Computer Vision and Image Understanding*, 78:138–156.

Trajković, M., et al. (1998). Fast corner detection. *Image and vision computing*, 16(2):75–87.

Triggs, B. (1997). Autocalibration and the absolute quadric. In *IEEE Computer Society CVPR Conference*, pages 609–614, San Juan, Pr.

Triggs, B., et al. (1999). Bundle Adjustment — A Modern Synthesis (from Photogrammetry to Computer Vision). *Lecture Notes in Computer Vision*, 1883:298–372.

Tuytelaars, T., et al. (2007). Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280.

Ueng, W.-D., et al. (1998). A sweep-surface fitting algorithm for reverse engineering. *Computers in Industry*, 35(3):261–273.

Uijlings, J. R. R., et al. (2013). Selective Search for Object Recognition. *International Journal of Computer Vision*, 104:154–171.

Ullman, S. (1979). The interpretation of structure from motion. In *Proceedings of the Royal Society of London*, pages 405–426.

Unser, M. (1999). Splines: A Perfect Fit for Signal and Image Processing. *Ieee Signal Processing Magazine*, pages 22–38.

Utcke, S. (2003). Error-Bounds on Curvature Estimation. In Griffin, L. D., et al., editors, *Scale Space Methods in Computer Vision*, pages 657–666. Springer, Berlin, Heidelberg.

Varma, M., et al. (2003). Texture classification: Are filter banks necessary? In *IEEE CVPR*, volume 2, pages II—-691.

Varma, M., et al. (2009). A Statistical Approach to Material Classification Using Image Patch Exemplars. 31(11):2032–2047.

Vázquez-Méndez, M. E., et al. (2016). The clothoid computation: a simple and efficient numerical algorithm. *Journal of Surveying Engineering*, 142(3):04016005.

Vincent, L., et al. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE TPAMI*, (6):583–598.

Volk, R., et al. (2014). Building Information Modeling (BIM) for existing buildings — Literature review and future needs. *Automation in Construction*, 38:109–127.

Vosselman, G., et al. (2004). Recognising Structures in Laser Scanning Point Clouds. *ISPRS Archives*, 46(8):33–38.

Walton, D. J., et al. (2005). A controlled clothoid spline. *Computers and Graphics (Pergamon)*, 29(3):353–363.

Wang, J., et al. (2012). VCells : Simple and Efficient Superpixels Using Edge-Weighted Centroidal Voronoi Tessellations. *IEEE TPAMI*, 34(6):1241–1247.

Wang, J., et al. (2008). A new calibration model of camera lens distortion. *Pattern Recognition*, 41(2):607–615.

Wang, W., et al. (2006). Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Transactions on Graphics*, 25(2):214–238.

Wang, Z., et al. (2011). Local intensity order pattern for feature description. In *IEEE ICCV*, pages 603–610.

Warren, S. G. (1982). Optical Properties of Snow. *Reviews of Geophysics and Space Physics*, 20.

Weigel, H., et al. (2006). Accurate Road Geometry Estimation for a Safe Speed Application. In *2006 IEEE Intelligent Vehicles Symposium*, pages 516–521.

Weinmann, M. (2016). *Reconstruction and Analysis of 3D Scenes*. Springer.

Wellner, P. D. (1993). Adaptive thresholding for the digitaldesk. *Xerox, EPC1993-110*, pages 1–19.

Wenzel, K. (2013). SURE - The ifp Software for Dense Image matchings.

Westoby, M. J., et al. (2012). 'Structure-from-Motion' photogrammetry : A low-cost , effective tool for geoscience applications. *Geomorphology*, 179:300–314.

Wikipedia (2018). *List of bobsleigh, luge, and skeleton tracks*. https://en.wikipedia.org/wiki/List of bobsleigh, luge, and skeleton tracks.

Williams, E. (2014). *Simultaneous Localisation and Mapping for Surveying Applications*. PhD thesis, University of Nottingham.

Wolfgang, F., et al. (2009). Detecting Interpretable and Accurate Scale-Invariant Keypoints. In *12th IEEE ICCV*.

Wu, C. (2011). VisualSFM : A Visual Structure from Motion System. *http://ccwu.me/vsfm/*.

Wu, C. (2014). Critical configurations for radial distortion self-calibration. *CVPR*, pages 25 – 32.

Wu, C., et al. (2012). Schematic surface reconstruction. *2012 IEEE CVPR Conference*, pages 1498–1505.

Xiong, H., et al. (2013a). A Study of Point Cloud Registration with Probability Product Kernel Functions. *2013 International Conference on 3d Vision (3dv 2013)*, pages 207–214.

Xiong, X., et al. (2013b). Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction*, 31:325–337.

XVI Olympic Winter Games Organising Committee (1992). Official Report of the XVI Olympic Winter Games. Technical report.

XVII Olympic Winter Games (1994). XVII Olympic Winter Games: Official Report, The Organising Committee. Technical report.

XVIII Olympic Winter Games (1998). Sixteen Days of Glory, The Organising Committee. Technical report.

Yin, K., et al. (2014). Morfit : Interactive Surface Reconstruction from Incomplete Point Clouds with Curve-Driven Topology and Geometry Control. *ACM Transactions on Graphics (TOG)*, 33(6):202:1—-202:12.

Yong, X., et al. (2015). *Descriptor evaluation and feature regression for multimodal image analysis*, volume 26. Springer.

Zhang, Y. L., et al. (1995). Optimum Control of Bodsled Steering. *Journal of Optimization Theory and Applications*, 85(1):1–19.

Zhu, Z., et al. (2010). Automation in Construction Parameter optimization for automated concrete detection in image data. *Automation in Construction*, 19(7):944–953.

Zuliani, M. (2012). RANSAC for Dummies.

Zuriaga, A., et al. (2010). Modeling Operating Speed and Deceleration on Two-Lane Rural Roads with Global Positioning System Data. *Transportation Research Record*, 2171:11–20.

Part VI

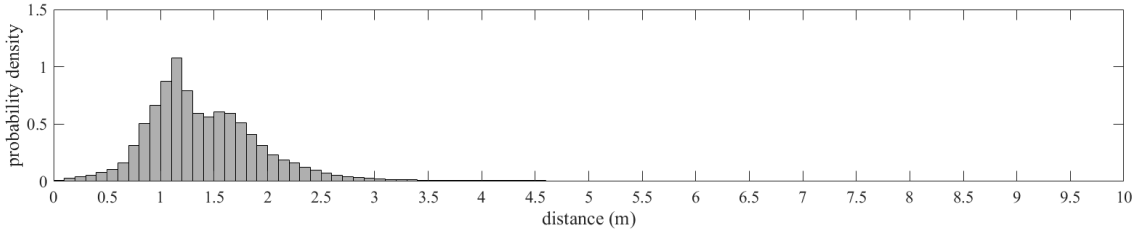
Appendices

Appendix A

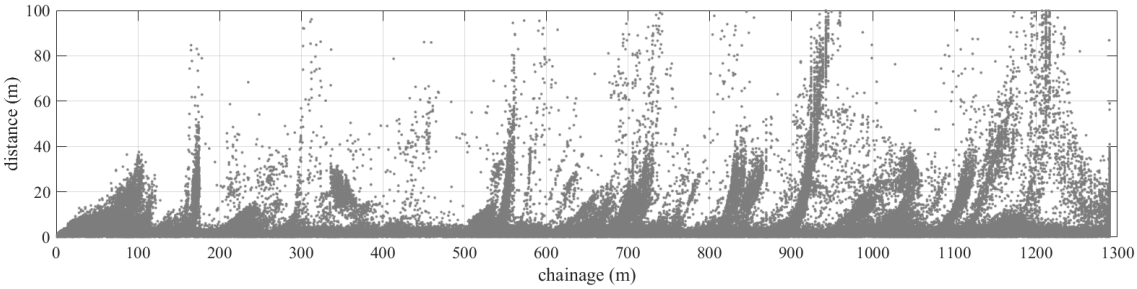
Preliminary Trials



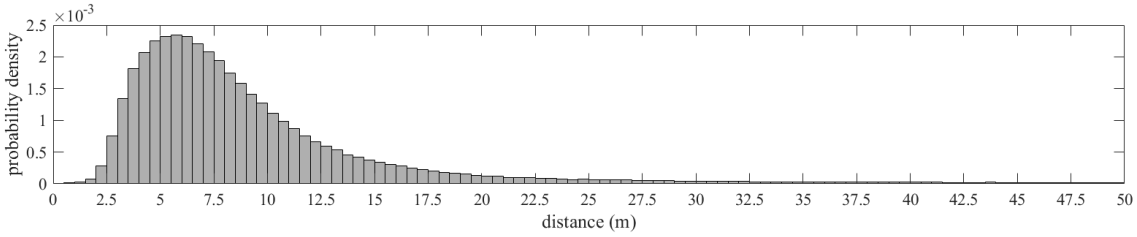
(a) Top view of the sparse point cloud.



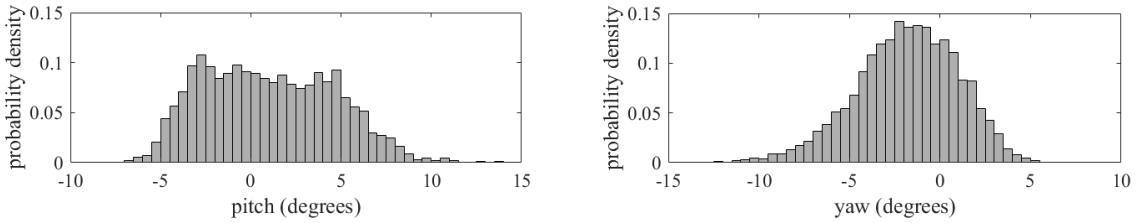
(b) Histogram of the orthogonal distance between the sensor path and each point.



(c) Scatter plot of orthogonal distances between the sensor path and each point with respect to its projection along the track.



(d) Histogram of the distances between each point and the camera it was detected on.

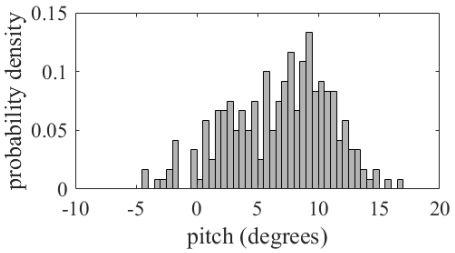


(e) Histogram of the camera rotation angles with respect to the sensor's trajectory.

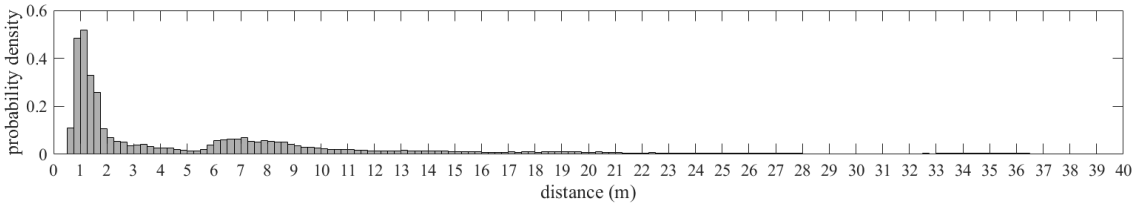
Figure A.1: Geometric properties of the imaging configuration used for the reconstruction of the Whistler track and of the generated sparse point cloud, q.v. section 5.1.



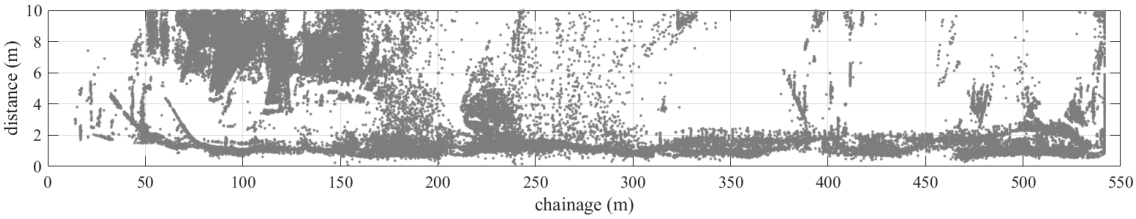
(a) Top view of the sparse point cloud.



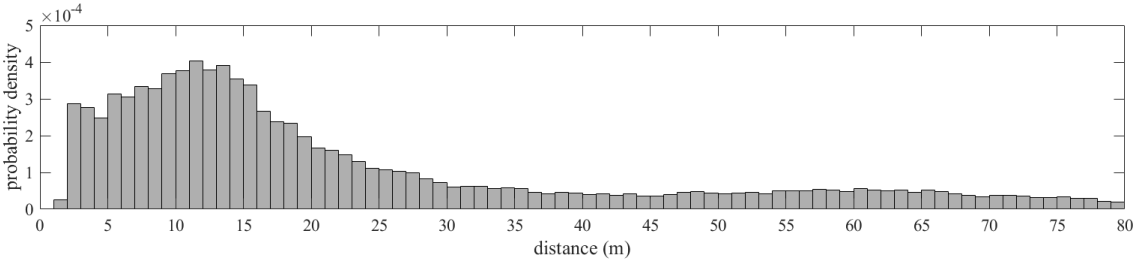
(b) Histogram of the camera pitch with respect to the sensor's trajectory.



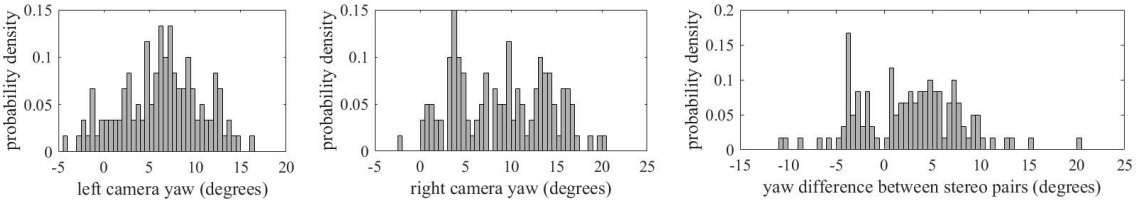
(c) Histogram of the orthogonal distance between the sensor path and each point.



(d) Orthogonal distances between the sensor path and each point with respect to its projection along the track.



(e) Histogram of the distances between each point and the camera it was detected on.



(f) Histogram of the camera yaw with respect to the sensor's trajectory.

Figure A.2: Geometric properties of the imaging configuration used for the reconstruction of a segment of the Winterberg track and of the generated sparse point cloud, q.v. section 5.2.

Appendix B

Network Design Simulation Scatter Plots

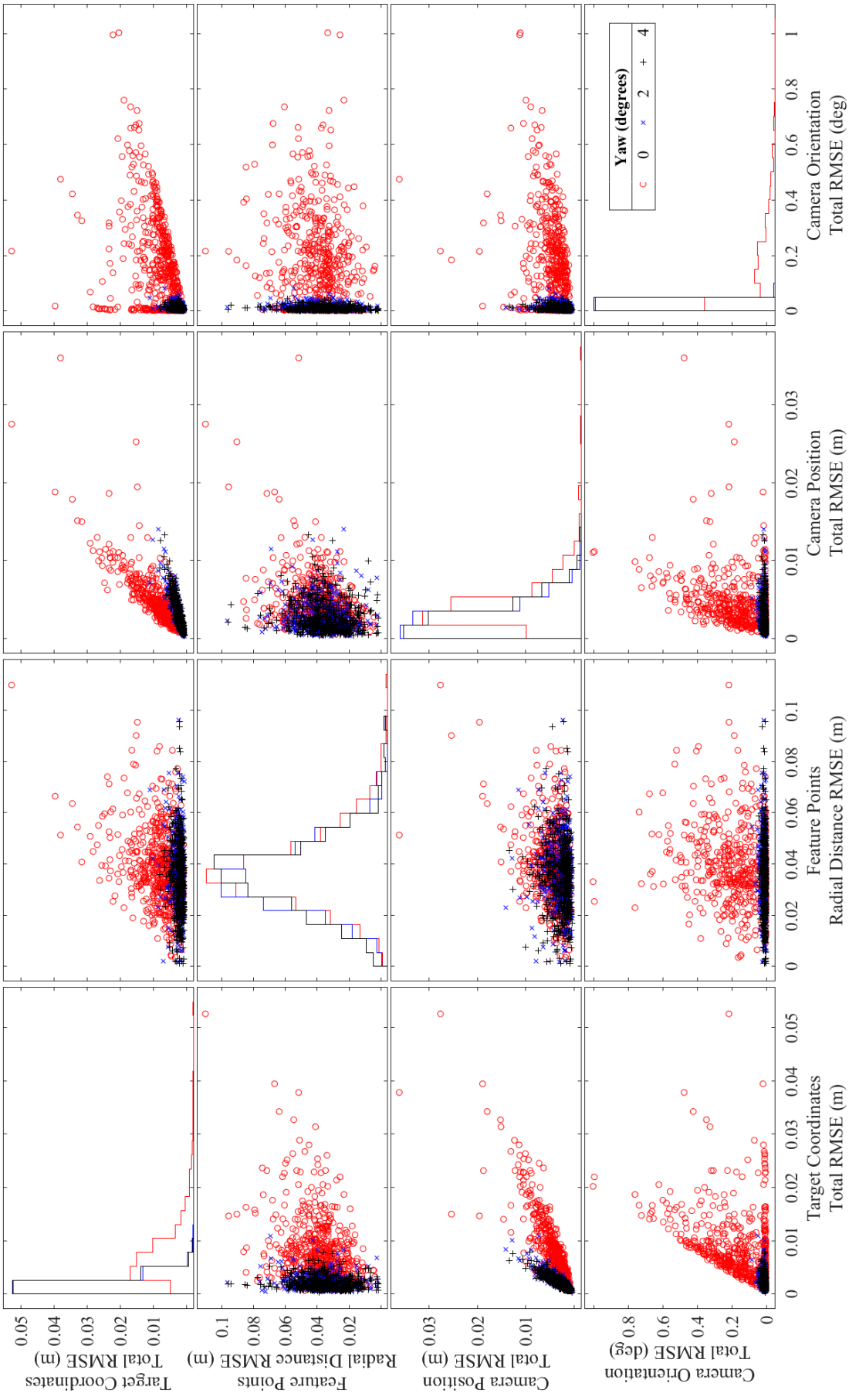


Figure B.1: Scatter plots and histograms of the object-space quality and camera exterior orientation errors as a result of the self-calibrating bundle adjustment, grouped according to the camera yaw.

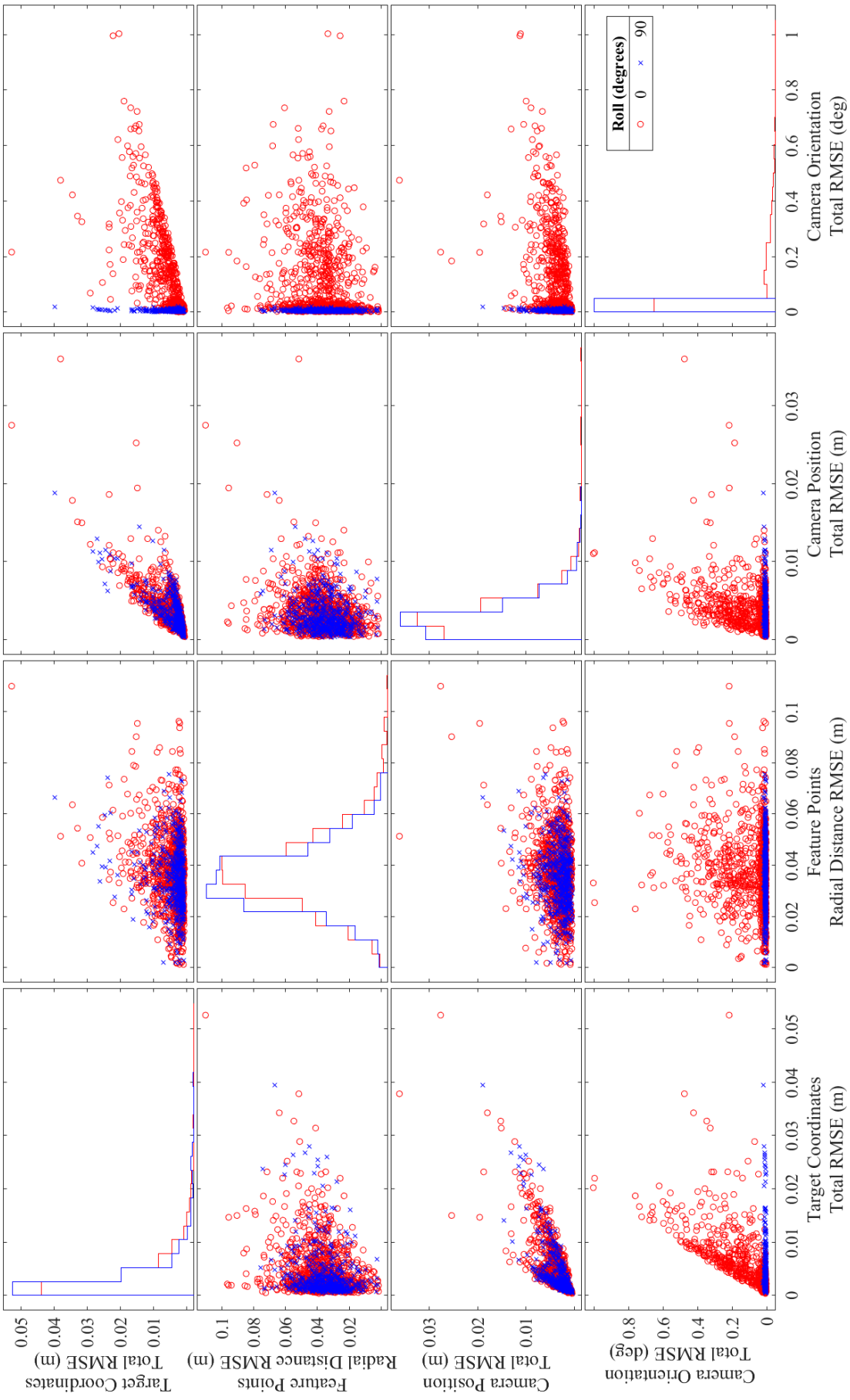


Figure B.2: Scatter plots and histograms of the object-space quality and camera exterior orientation errors as a result of the self-calibrating bundle adjustment, grouped according to whether or not cameras with varying roll angle are included in the network.

Appendix C

Image Sequence Quality Reports

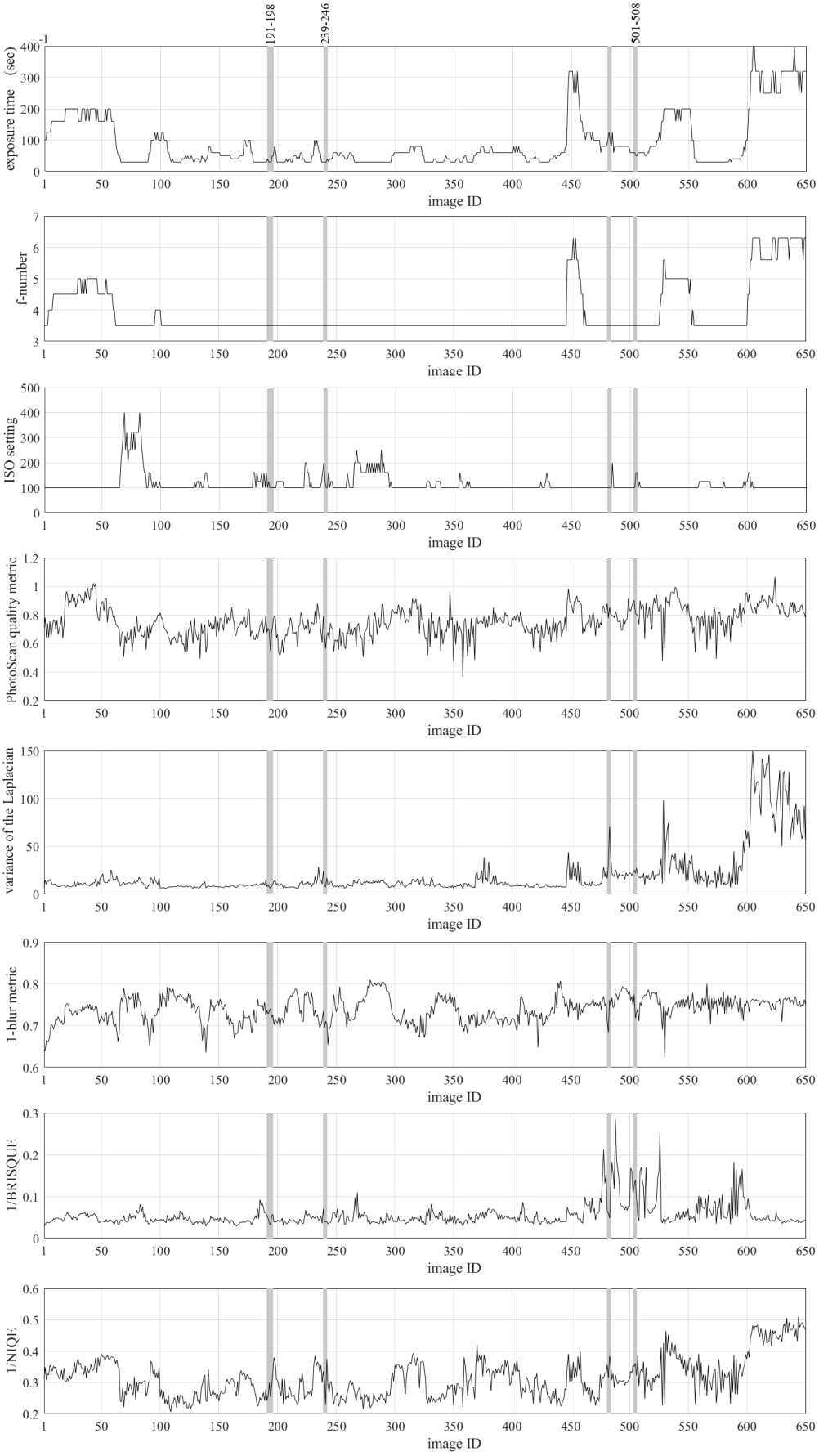


Figure C.1: The camera settings and image quality metrics for the Winterberg case study. The grey areas indicate problems in the relative image orientation process

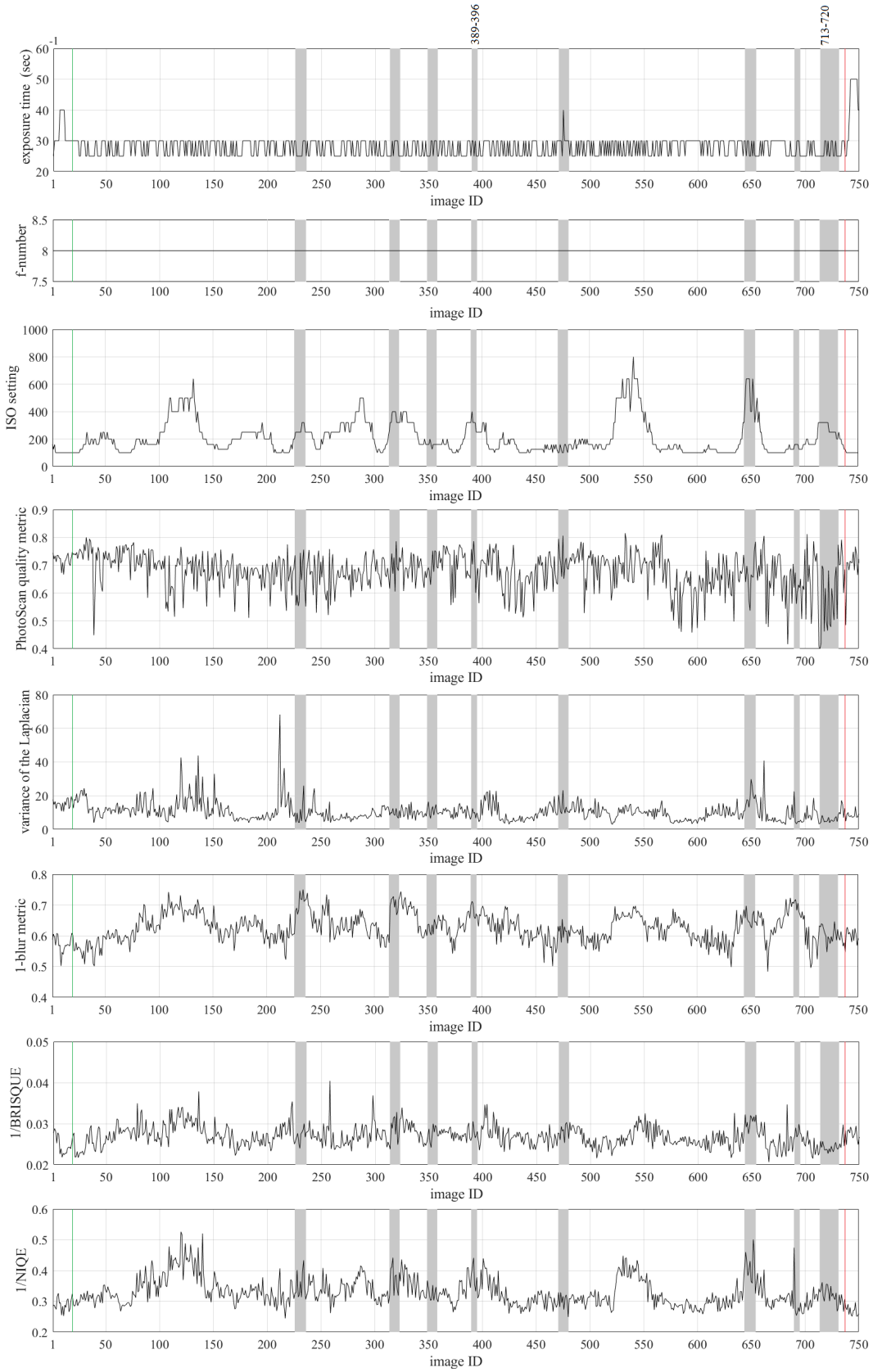


Figure C.2: The camera settings and image quality metrics for the Lillehammer case study. The grey areas indicate problems in the relative image orientation process.

Table C.1: Image sequence 191–198 along the Winterberg track.

			
Quality: 0.68	Reflectance: Poor	Base: 0.55 m	Quality: 0.71	Reflectance: Poor
Interval: 6.0 m				
			
Quality: 0.68	Reflectance: Poor	Base: 0.60 m	Quality: 0.55	Reflectance: Poor
Interval: 6.5 m				
			
Quality: 0.68	Reflectance: Average	Base: 0.55 m	Quality: 0.74	Reflectance: Average
Interval: 6.5 m				
			
Quality: 0.79	Reflectance: Average	Base: 0.70 m	Quality: 0.8	Reflectance: Average
Alignment success rate at different image resolution levels				
Low: 0% Medium: 20% High: 66%				

Table C.2: Image sequence 201–208 along the Winterberg track.

	Base: 0.55 m	
Quality: 0.53 Reflectance: Poor		Quality: 0.51 Reflectance: Poor
Interval: 3.5 m		
	Base: 0.55 m	
Quality: 0.55 Reflectance: Average		Quality: 0.62 Reflectance: Poor
Interval: 4.0 m		
	Base: 0.55 m	
Quality: 0.53 Reflectance: Average		Quality: 0.62 Reflectance: Average
Interval: 4.0 m		
	Base: 0.60 m	
Quality: 0.66 Reflectance: Good		Quality: 0.65 Reflectance: Average
Alignment success rate at different image resolution levels		
Low: 96% Medium: 94% High: 100%		

Table C.3: Image sequence 239–246 along the Winterberg track.

	Base: 0.60 m	
Quality: 0.79 Reflectance: Average		Quality: 0.65 Reflectance: Average
Interval: 6.5 m		
	Base: 0.55 m	
Quality: 0.56 Reflectance: Poor		Quality: 0.65 Reflectance: Poor
Interval: 4.0 m		
	Base: 0.55 m	
Quality: 0.68 Reflectance: Poor		Quality: 0.62 Reflectance: Average
Interval: 4.0 m		
	Base: 0.55 m	
Quality: 0.74 Reflectance: Average		Quality: 0.71 Reflectance: Poor

Alignment success rate at different image resolution levels

Low: 0% Medium: 0% High: 0%

Table C.4: Image sequence 501–508 along the Winterberg track.

	Base: 0.55 m	
Quality: 0.83 Reflectance: Good		Quality: 0.86 Reflectance: Good
Interval: 5.5 m		
	Base: 0.55 m	
Quality: 0.89 Reflectance: Average		Quality: 0.90 Reflectance: Average
Interval: 7.5 m		
	Base: 0.60 m	
Quality: 0.81 Reflectance: Poor		Quality: 0.80 Reflectance: Poor
Interval: 7.0 m		
	Base: 0.65 m	
Quality: 0.80 Reflectance: Average		Quality: 0.73 Reflectance: Average
Alignment success rate at different image resolution levels		
Low: 56% Medium: 100% High: 4%		

Table C.5: Image sequence 105–112 along the Lillehammer track.

	Base: 0.75 m	
Quality: 0.73 Reflectance: Average		Quality: 0.54 Reflectance: Average
	Interval: 2.40 m	
	Base: 0.70 m	
Quality: 0.57 Reflectance: Poor		Quality: 0.53 Reflectance: Poor
	Interval: 2.60 m	
	Base: 0.75 m	
Quality: 0.73 Reflectance: Very Poor		Quality: 0.59 Reflectance: Very Poor
	Interval: 2.80 m	
	Base: 0.70 m	
Quality: 0.62 Reflectance: Very Poor		Quality: 0.62 Reflectance: Very Poor

Alignment success rate at different image resolution levels

Low: 42% Medium: 54% High: 74%

Table C.6: Image sequence 389–396 along the Lillehammer track.

		Base: 0.80 m	
Quality: 0.65	Reflectance: Very Poor		Quality: 0.76 Reflectance: Very Poor
		Interval: 4.30 m	
		Base: 0.80 m	
Quality: 0.70	Reflectance: Very Poor		Quality: 0.79 Reflectance: Very Poor
		Interval: 4.10 m	
		Base: 0.80 m	
Quality: 0.74	Reflectance: Very Poor		Quality: 0.60 Reflectance: Very Poor
		Interval: 4.00 m	
		Base: 0.70 m	
Quality: 0.66	Reflectance: Very Poor		Quality: 0.69 Reflectance: Very Poor
Alignment success rate at different image resolution levels			
Low: 0% Medium: 0% High: 0%			

Table C.7: Image sequence 581–588 along the Lillehammer track.

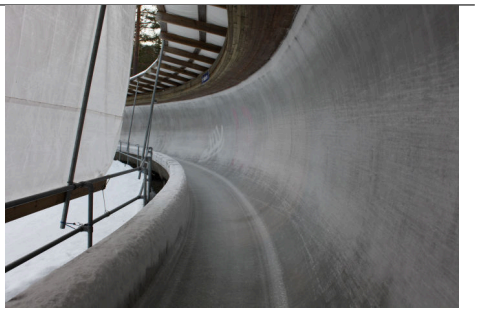
	Base: 0.60 m	
Quality: 0.69 Reflectance: Good		Quality: 0.53 Reflectance: Good
Interval: 3.90 m		
	Base: 0.65 m	
Quality: 0.47 Reflectance: Good		Quality: 0.64 Reflectance: Good
Interval: 3.70 m		
	Base: 0.65 m	
Quality: 0.46 Reflectance: Good		Quality: 0.59 Reflectance: Good
Interval: 3.30 m		
	Base: 0.75 m	
Quality: 0.61 Reflectance: Good		Quality: 0.65 Reflectance: Good
Alignment success rate at different image resolution levels		
Low: 100% Medium: 100% High: 100%		

Table C.8: Image sequence 713–720 along the Lillehammer track.

	Base: 0.90 m	
Quality: 0.41 Reflectance: Average		Quality: 0.40 Reflectance: Average
	Interval: 6.0 m	
	Base: 0.80 m	
Quality: 0.42 Reflectance: Poor		Quality: 0.70 Reflectance: Poor
	Interval: 6.1 m	
	Base: 0.90 m	
Quality: 0.69 Reflectance: Very Poor		Quality: 0.46 Reflectance: Very Poor
	Interval: 5.8 m	
	Base: 0.90 m	
Quality: 0.62 Reflectance: Very Poor		Quality: 0.50 Reflectance: Very Poor
Alignment success rate at different image resolution levels		
Low: 2% Medium: 72% High: 58%		

Appendix D

Camera Calibration

The initial camera calibration was performed in ©Photometrix Australis (2007) software with images of a three-dimensional frame with reflective coded targets, q.v. figure D.1, and subsequently transformed to the ©Agisoft PhotoScan (2018) file format.

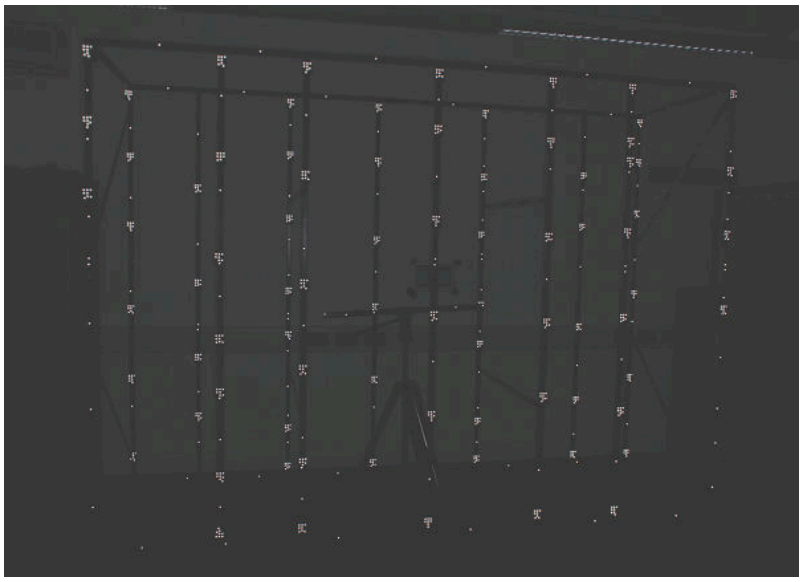


Figure D.1: One of the images of the 3D reflective target array acquired using a ring flash and used for the camera calibration.

Table D.1: Calibration Report 1 for the Canon EOS5D Mark II DSLR camera with a 28 mm lens

Adjustment:	Free-Network	
Number of Points:	768	
Number of Images:	25	
RMS of Image coordinates:	0.54 µm	
Sensor Size	Horizontal	Vertical
pixels	5616	3744
mm	36	24
Pixel Size (mm)	0.0064	0.0064

Parameter	Australis		PhotoScan
	Value	Standard Error	Value
c	28.4890	3.226e-004 mm	4451.5415 pixels
x_{pp}	-0.0381	3.249e-004 mm	-5.9546 pixels
y_{pp}	-0.0424	2.807e-004 mm	6.6254 pixels
K_1[a]	1.32261e-004	8.298e-008	-0.10820493
K_2	-1.97946e-007	4.235e-010	0.17156102
K_3	-6.36082e-011	7.036e-013	-0.10704497
P_1	-3.57233e-006	1.214e-007	9.734e-005
P_2	-2.23034e-006	1.031e-007	-6.063e-005

[a] Maximum observational radial distance encountered: 20.95 mm

Table D.2: Calibration Report 2 for the Canon EOS5D Mark II camera with a 28 mm lens

Adjustment:	Free-Network	
Number of Points:	806	
Number of Images:	36	
RMS of Image coordinates:	0.48 µm	
Sensor Size	Horizontal	Vertical
pixels	5616	3744
mm	36	24
Pixel Size (mm)	0.0064	0.0064

Parameter	Australis		PhotoScan
	Value	Standard Error	Value
c	28.4663	2.464e-004 mm	4447.9737 pixels
x_{pp}	0.0040	2.953e-004 mm	0.6243 pixels
y_{pp}	-0.0063	2.475e-004 mm	0.9857 pixels
K_1[a]	1.33482e-004	6.034e-008	-0.10886647
K_2	-2.06844e-007	3.305e-010	0.17589854
K_3	-5.33127e-011	5.762e-013	-0.11115098
P_1	-1.38918e-006	1.047e-007	3.785e-005
P_2	-6.41515e-006	8.731e-008	-0.00017439

[a] Maximum observational radial distance encountered: 20.72 mm

Table D.3: Calibration Report for the Nikon D300 DSLR camera with a 28 mm lens

Adjustment:	Free-Network	
Number of Points:	750	
Number of Images:	28	
RMS of Image coordinates:	0.35 μm	
Sensor Size	Horizontal	Vertical
pixels	4288	2848
mm	24	16
Pixel Size (mm)	0.0055	0.0055

Parameter	Australis		PhotoScan
	Value	Standard Error	Value
c	28.6937	2.757e-004 mm	5217.0300 pixels
x_{pp}	0.1118	3.794e-004 mm	20.3100 pixels
y_{pp}	-0.0024	3.473e-004 mm	0.4219 pixels
K_1[a]	1.49615e-004	1.279e-007	-0.12308177
K_2	-1.73907e-007	1.541e-09	0.16143252
K_3	-3.66460e-011	5.589e-012	-0.09988380
P_1	2.51153e-005	1.188e-007	-0.00066364
P_2	-2.51740e-005	1.091e-007	-0.00066941

[a] Maximum observational radial distance encountered: 13.99 mm

Appendix E

GNSS RTK Survey Reports

A survey-grade Leica Viva GS10 GNSS receiver combined with and AS10 compact antenna, with a nominal horizontal and vertical accuracy of $8mm \pm 1ppm$ and $15mm \pm 1ppm$, respectively, was used for the GNSS surveys using the single baseline real time kinematic technique. The measured coordinates were transformed from WGS84 to a local plane tangent to the reference ellipsoid at a principal point towards the centre of the site. Considering the accuracy of the EGM08 geoid model (Pavlis et al., 2012), the geoid height at the entire site was assumed to be equal to that at the principal point.

Table E.1: Lillehammer Track Ground Control Point Coordinates (WGS84).

Point ID	X (m)	Y (m)	Z (m)	Position Quality (m)
GCP1	3028119.343	556522.486	5567525.313	0.014
GCP2	3028101.826	556532.005	5567531.223	0.013
GCP3	3028066.500	556552.042	5567542.437	0.019
GCP4	3028032.418	556562.482	5567553.806	0.017
GCP5	3027996.402	556578.812	5567564.737	0.012
GCP6	3027989.149	556638.720	5567550.174	0.013
GCP7	3028007.437	556686.775	5567528.672	0.018
GCP8	3027984.700	556707.992	5567535.543	0.018
GCP9	3027958.291	556727.029	5567544.662	0.013
GCP10	3027920.132	556766.813	5567551.797	0.045
GCP11	3027928.720	556763.910	5567547.145	0.019
GCP12	3027935.592	556765.565	5567543.110	0.013
GCP13[a]	3028003.398	556765.548	5567499.281	0.014
GCP14	3028081.199	556771.384	5567443.899	0.022
GCP15	3028082.548	556825.119	5567427.080	0.019
GCP16	3027994.515	557002.006	5567420.842	0.023
GCP17	3027981.597	556957.306	5567441.342	0.015

[a] the origin of the local coordinate system

Table E.2: Igls Track Ground Control Point Coordinates (WGS84).

Point ID	X (m)	Y (m)	Z (m)	Position Quality (m)
GCP1	4254290.717	860528.311	4659335.167	0.025
GCP2	4254297.169	860512.069	4659332.282	0.013
GCP3	4254305.085	860486.674	4659324.496	0.033
GCP4	4254314.757	860464.034	4659316.990	0.015
GCP5	4254316.752	860430.266	4659316.166	0.019
GCP6	4254254.170	860434.331	4659354.327	0.019
GCP7	4254286.152	860290.330	4659330.092	0.027
GCP8	4254288.303	860263.456	4659329.549	0.013
GCP9	4254284.315	860239.349	4659333.176	0.032
GCP10[a]	4254224.740	860275.888	4659357.392	0.028
GCP11	4254112.525	860279.603	4659435.338	0.024
GCP12	4254123.691	860248.837	4659426.024	0.022
GCP13	4254132.439	860193.555	4659421.573	0.024
GCP14	4254140.004	860158.818	4659417.478	0.018
GCP15	4254157.350	860076.013	4659408.836	0.013
GCP16	4254204.051	860041.109	4659382.082	0.019
GCP17	4254234.423	860037.406	4659362.007	0.028
GCP18	4254270.472	860036.224	4659339.186	0.035

[a] the origin of the local coordinate system

Appendix F

Lillehammer Case Study

F.1 Design Plans Transformation Results

The coordinates (U, V) for every point i on each curve were calculated from the distorted coordinates (X, Y) using the below 2D second degree polynomial geometric transformation (Goshtasby, 1986, 1988), q.v. figure F.1:

for the horizontal alignment

$$U_i = -1247.981 + 1.185X_i + 0.292Y_i - 3.646e-06X_iY_i + 8.381e-06X_i^2 - 1.636e-05Y_i^2$$

$$V_i = 3837.750 + 0.0087X_i + 0.518Y_i - 1.311e-06X_iY_i + 1.897e-06X_i^2 + 4.138e-06Y_i^2$$

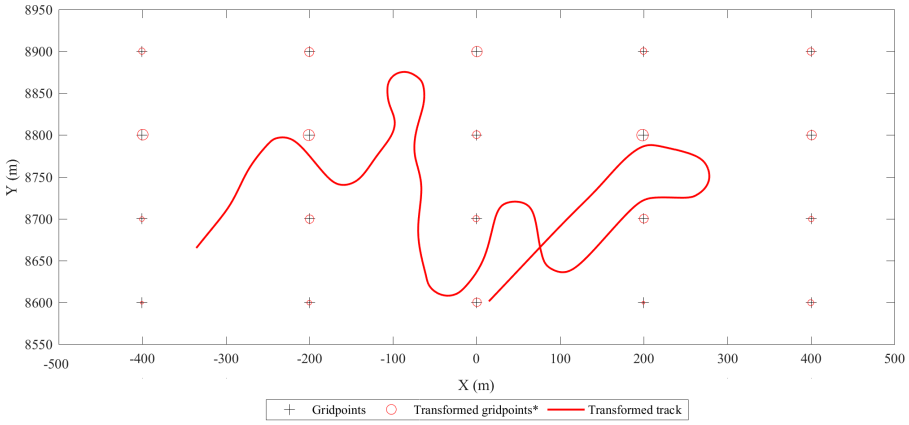
for the profile

$$U_i = -2.680 + 1.234X_i + 0.011Y_i + 1.691e-06X_iY_i - 4.373e-06X_i^2 - 5.535e-06Y_i^2$$

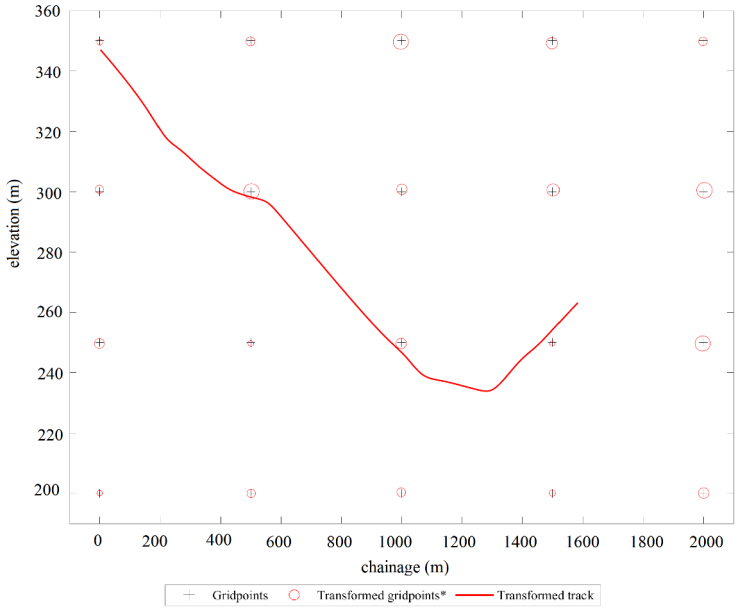
$$V_i = 166.515 - 0.00055X_i + 0.168Y_i + 2.370e-07X_iY_i + 1.824e-07X_i^2 - 3.793e-07Y_i^2$$

F.2 Zeb1 Data Sets

The point clouds and sensor trajectories for each laser scan survey using the Zeb1 are illustrated in figures F.3–F.8. The color map that indicates the quality, estimated from the feature-based registration of the successive 1-second data packets, and the acquisition time of each point is depicted in figure F.2.



(a) Horizontal alignment.



(b) Profile.

Figure F.1: Lillehammer design plans transformation. The radius of the red circles at each grid point of the raster input is relative the transformation error. A nonlinear distortion is to be expected along the first 200–400 m (the start line is on the left).

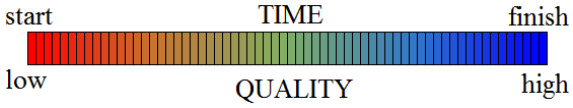
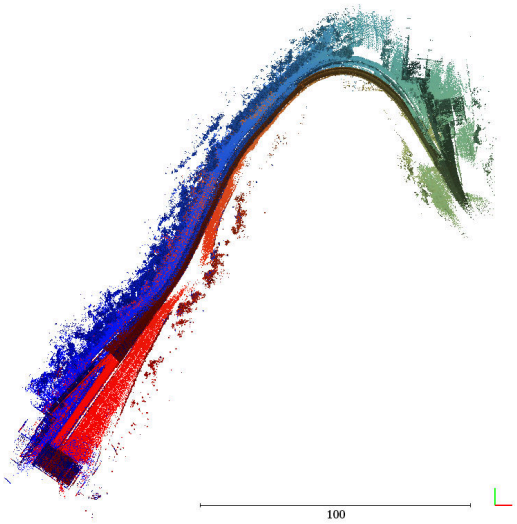
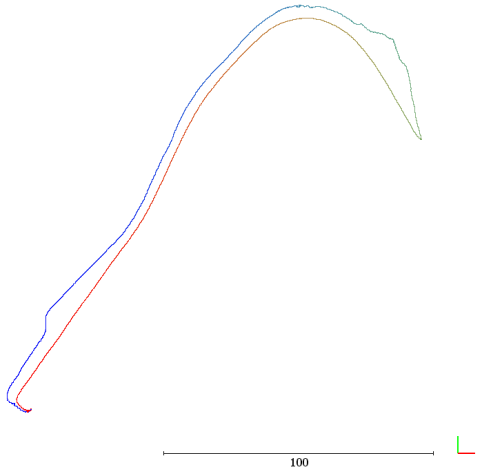


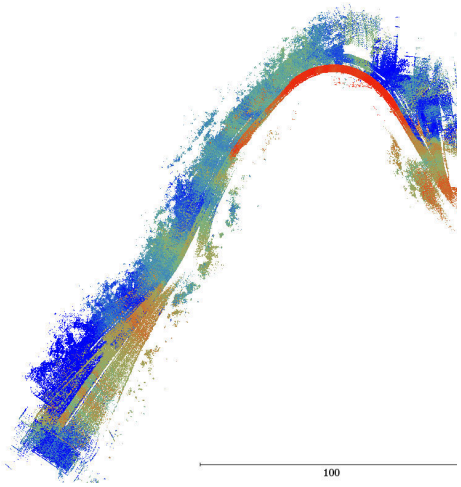
Figure F.2: Colour-coded legend for the Zeb1 point clouds. The color map indicates the quality, estimated from the feature-based registration of the successive 1-second data packets, and the acquisition time of each point.



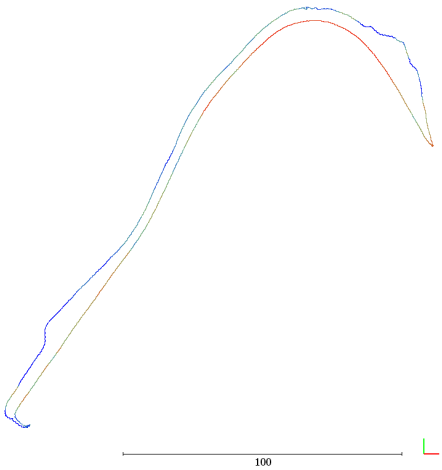
(a) Top view of the point cloud, colour-coded according to acquisition time.



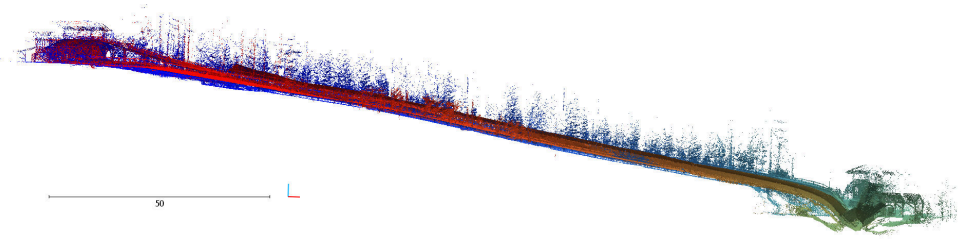
(b) Top view of the sensor path, colour-coded according to acquisition time.



(c) Top view of the point cloud colour-coded according to position estimation uncertainty.

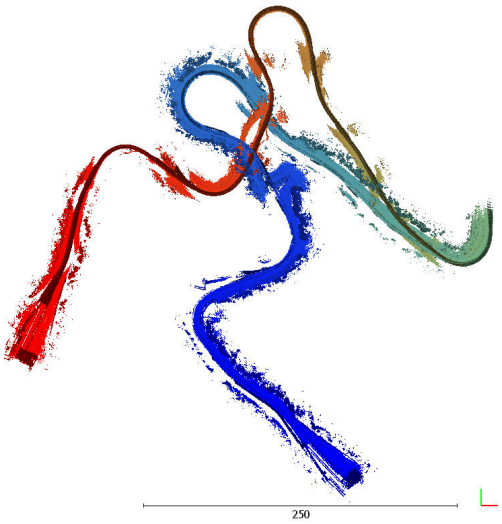


(d) Top view of the sensor path, colour-coded according to position estimation uncertainty.

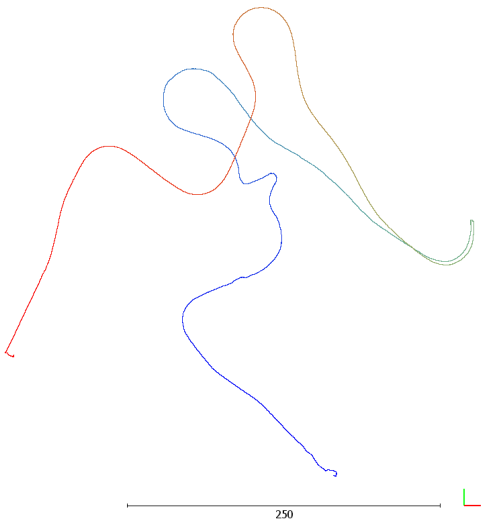


(e) Side view of the point cloud, colour-coded according to acquisition time.

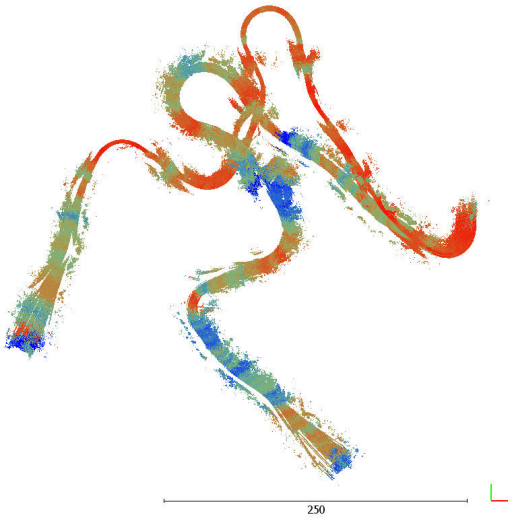
Figure F.3: *Zeb*1*s*1: Upper start - Lower start (closed loop).



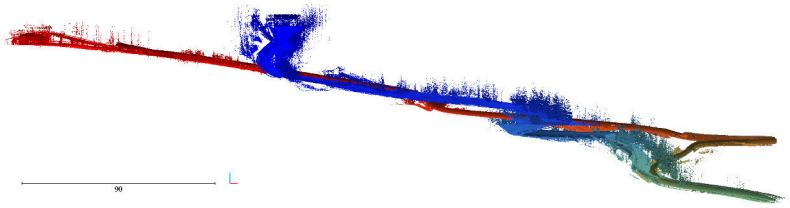
(a) Top view of the point cloud, colour-coded according to acquisition time.



(b) Top view of the sensor path, colour-coded according to acquisition time..



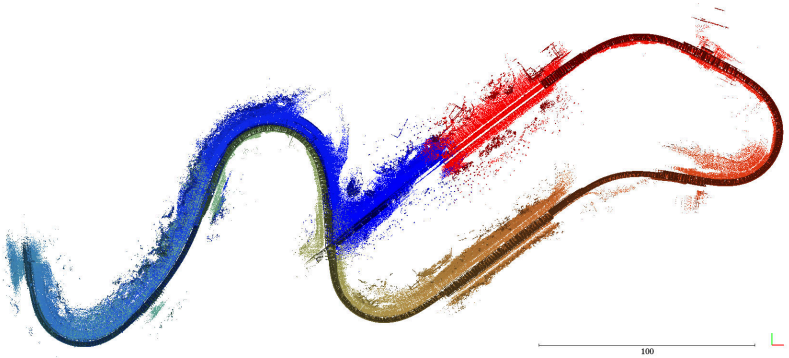
(c) Top view of the point cloud, colour-coded according to position estimation uncertainty.



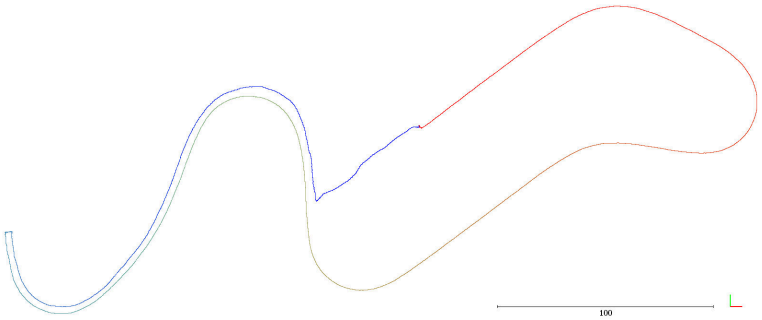
(d) Side view of the point cloud, colour-coded according to acquisition time.

Figure F.4: LilZebSmK11Sm.

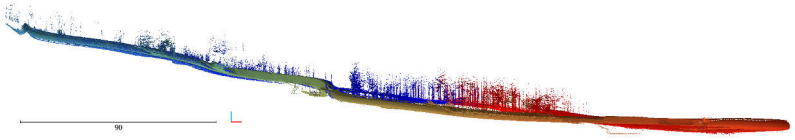
Figure F.5: *Zeb*1*s*2: Upper - Bend 11 (closed loop).



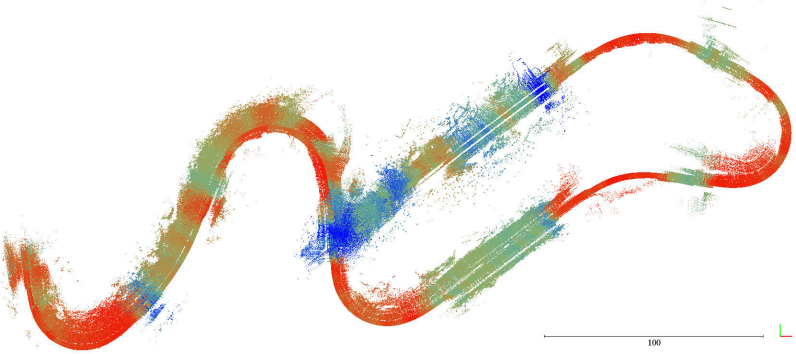
(a) Top view of point cloud, colour-coded according to acquisition time.



(b) Top view of the sensor path, colour-coded according to acquisition time.

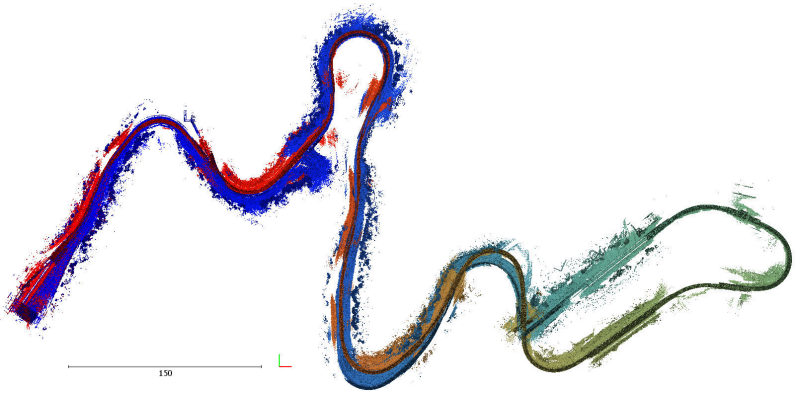


(c) Side view of the point cloud, colour-coded according to acquisition time.

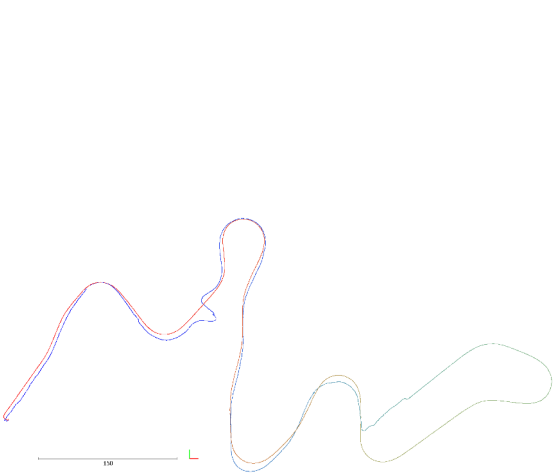


(d) Top view of the point cloud, colour-coded according to position estimation uncertainty.

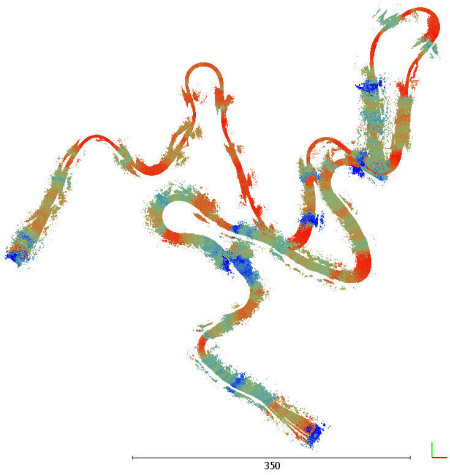
Figure F.6: Z$Zeb1s3$: Finish - Bend 11 (closed loop).



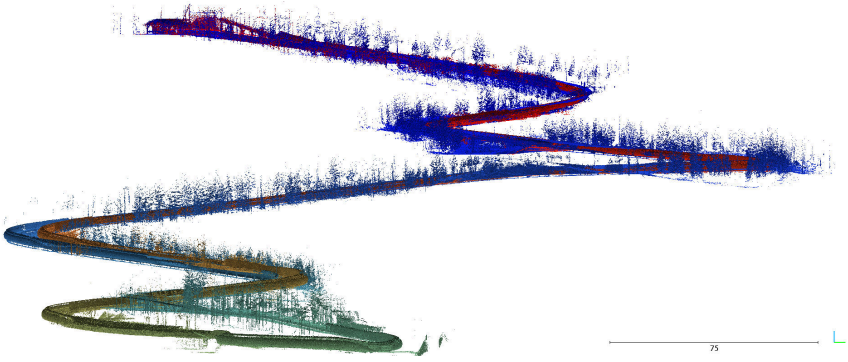
(a) Top view of the point cloud, colour-coded according to acquisition time.



(b) Top view of the sensor path, colour-coded according to acquisition time.

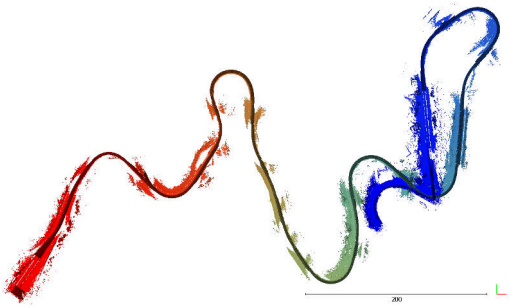


(c) Top view of the point cloud, before the loop closure, colour-coded according to position estimation uncertainty.

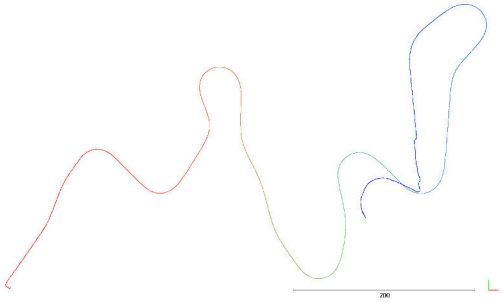


(d) Side view of the point cloud, colour-coded according to acquisition time.

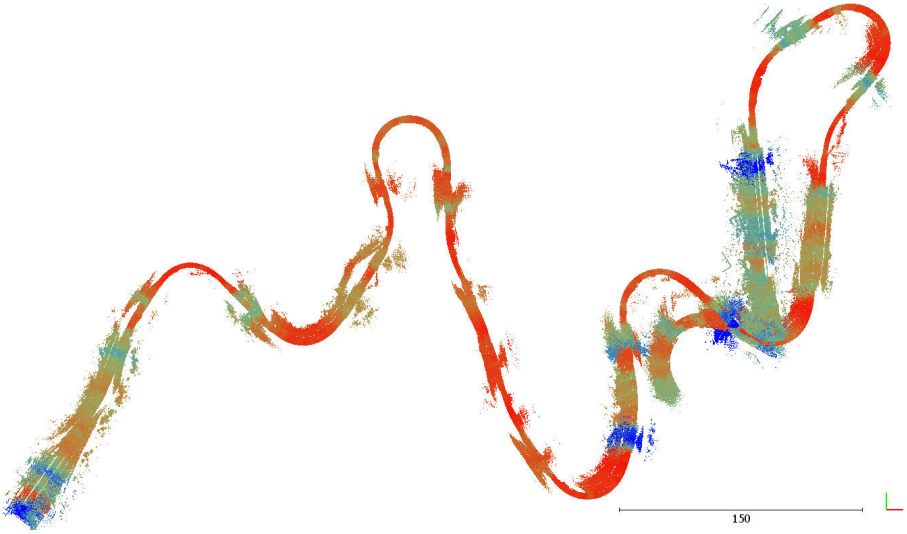
Figure F.7: *Zeb*1*s*4: Upper start - Finish - Upper start (closed loop).



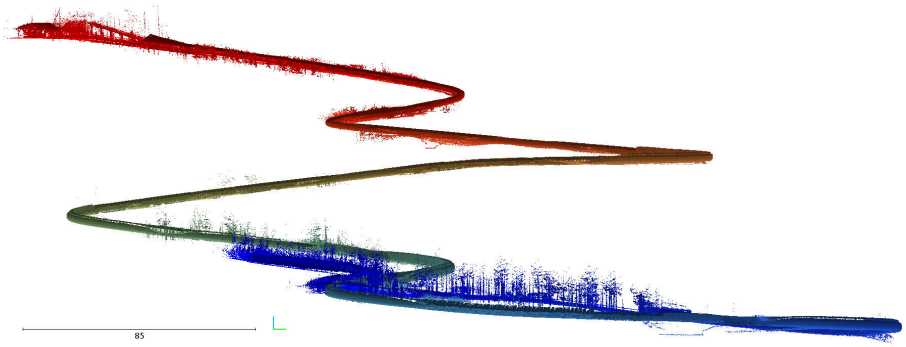
(a) Top view of the point cloud, colour-coded according to acquisition time.



(b) Top view of the sensor path colour-coded according to acquisition time.



(c) Top view of the point cloud, colour-coded according to position estimation uncertainty.



(d) Side view of the point cloud, colour-coded according to acquisition time.

Figure F.8: *Zeb*1*s*5: Upper start - Finish.

Appendix G

Igls Case Study

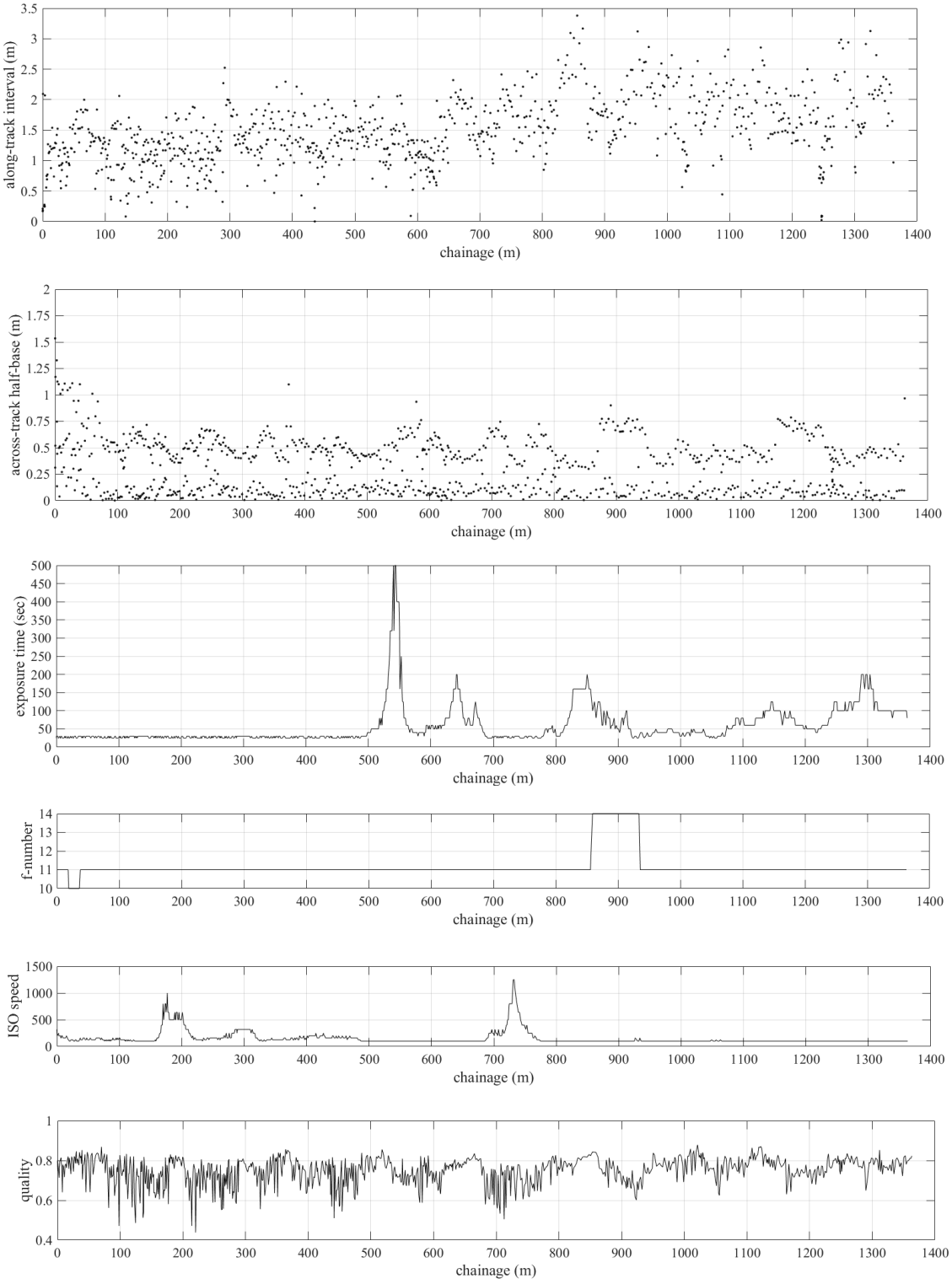


Figure G.1: The along- and across-track distances between the camera stations and the camera settings for data set *IglsConcC*5 that was collected at the Igls track at the concrete state.

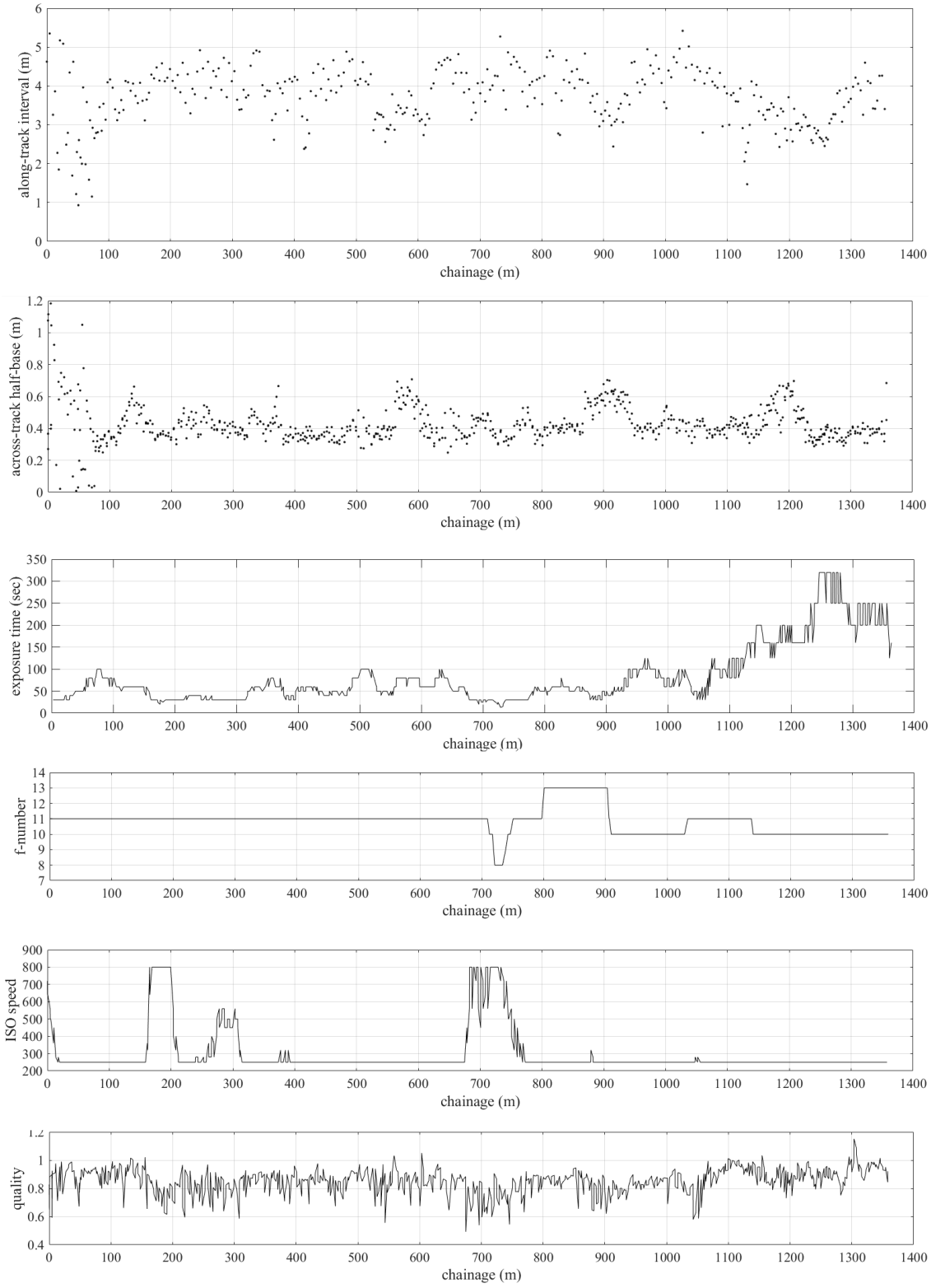
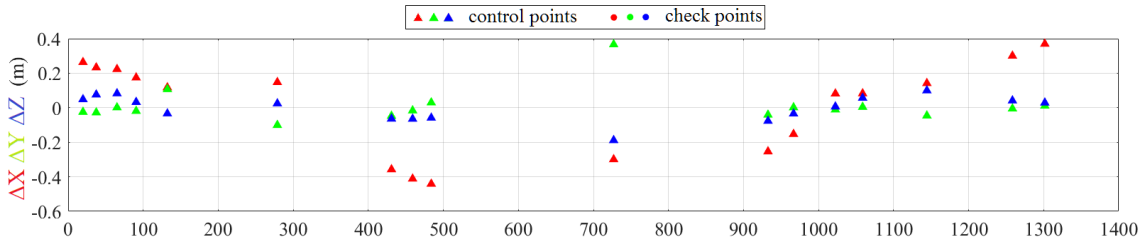


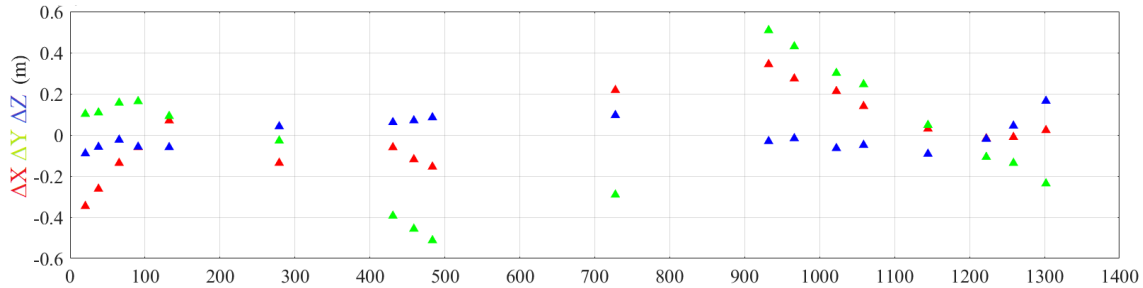
Figure G.2: The along- and across-track distances between the camera stations and the camera settings for data set *IglsConcN*11 that was collected at the Igls track at the concrete state.

Table G.1: Image orientation, dense matching, and mesh generation report for the Igls concrete track scan.

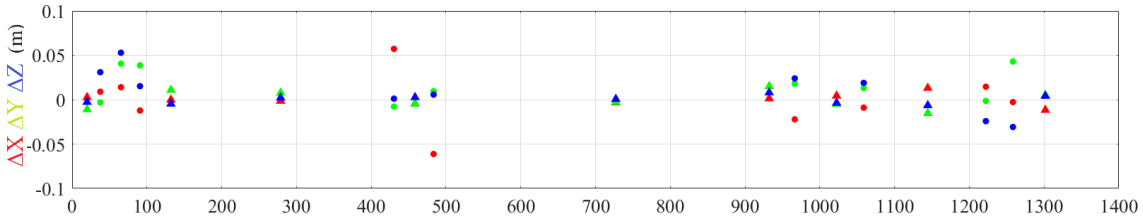
						*IglsConcC*5	*IglsConcN*11	*IglsConcCN*
		Image Orientation	Point Limit		images	936	730	1666
					resolution	x4	x4	x4
					detected features	40000	-	200000
					matched features	2000	-	4000
			Duration		feature matching	87	1440	1560
					image orientation	10	37	54
					sparse point cloud	449316	1227358	1600000
SfM	Solution	Similarity Transform.			RMSE (m)	0.35	0.29	0.32
					max (m)	0.6	0.51	0.66
					RMS reproj. (pix.)	0.62	0.52	0.51
		Bundle Adjustment	Control Check		RMSE (m)	0.02	0.03	0.03
					max	0.03	0.07	0.07
					RMSE (m)	0.05	0.04	0.04
					max (m)	0.07	0.05	0.05
					RMS reproj. (pix.)	0.62	0.52	0.47
		Dense Matching	Duration		resolution	1/16		1/64
					filtering	Aggressive		Aggressive
					bounding boxes	22		1
					depth maps	709		338
					point cloud	329		415
					point cloud points	138682129		23161579
		Mesh			duration	1080		29
					faces	4963000		5000000
					vertices	2566567		2524305



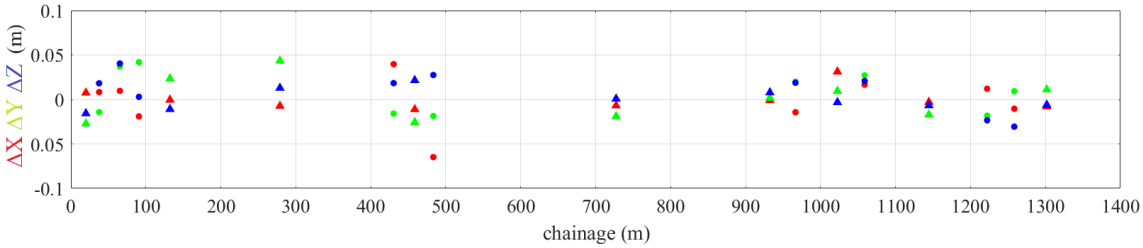
(a) Residuals of the similarity transformation of *IglsConcN*11 using all the available GCPs.



(b) Residuals of the similarity transformation of *IglsConcC*5 using all the available GCPs.



(c) Residuals of a fully constrained bundle adjustment of *IglsConcC*5 using GCPs 1, 5, 6, 8, 10, 11, 13, 15, and 18 as control points.



(d) Residuals of a fully constrained bundle adjustment of *IglsConcCN* using GCPs 1, 5, 6, 8, 10, 11, 13, 15, and 18 as control points.

Figure G.3: The control and check point coordinate residuals for the similarity transformation and constrained bundle adjustment of data sets *IglsConcC*5, *IglsConcN*11, and *IglsConcCN*.

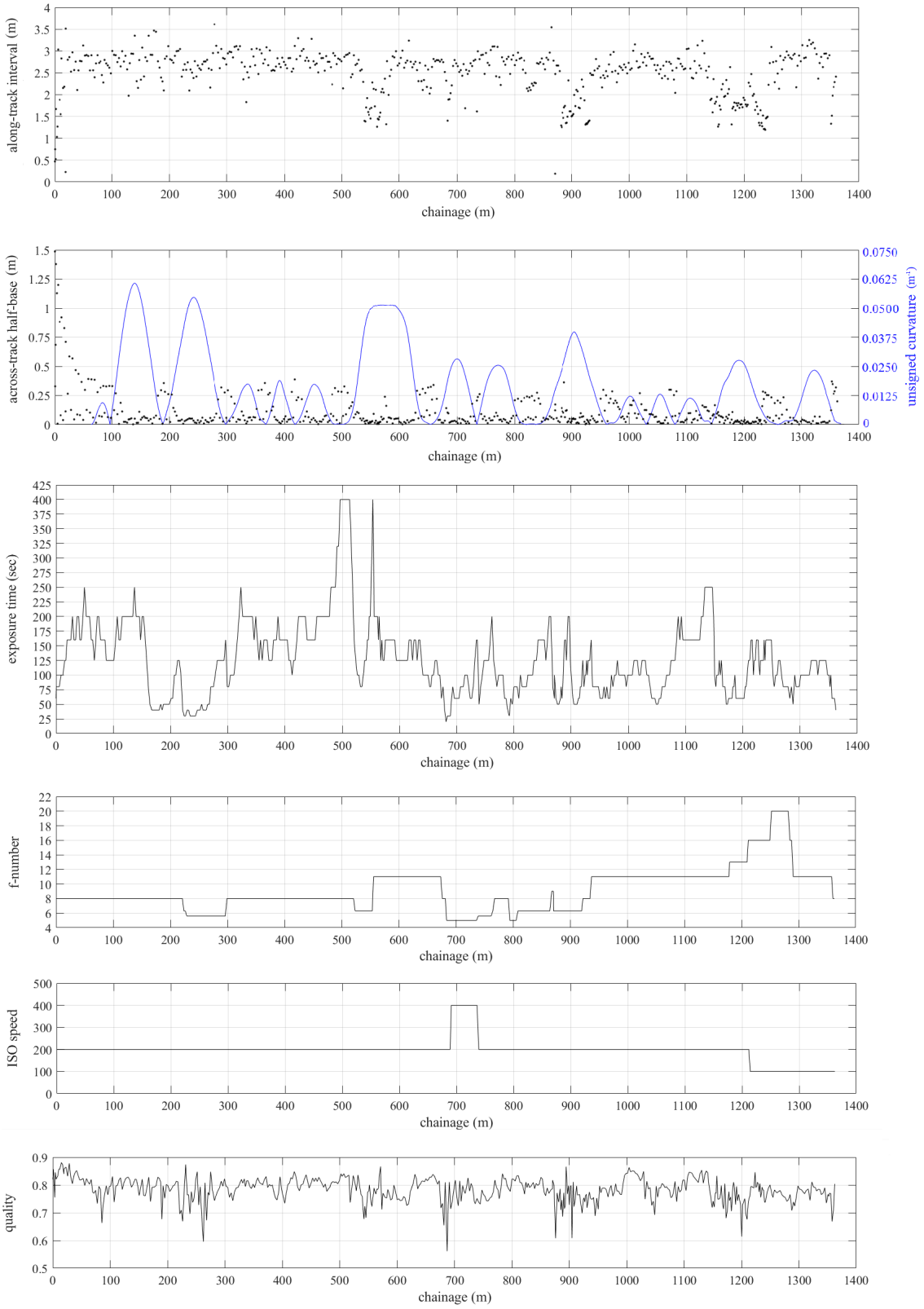
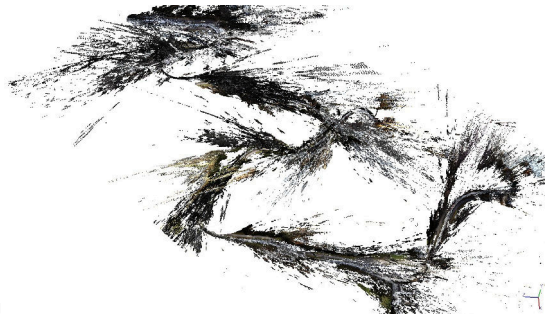


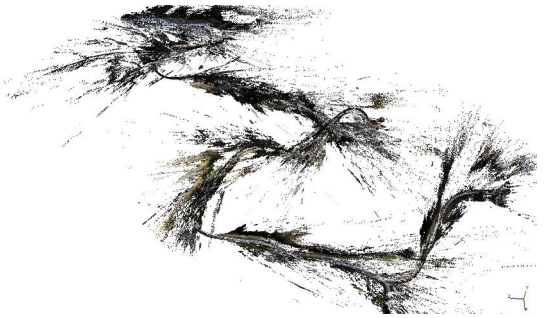
Figure G.4: The camera settings, across-track half-base, and along-track distances, for each camera station, with respect to its position for image set *IglsIce*26.



(a) Sparse point cloud 833520 points).



(b) Dense point cloud without masks.
(Resolution: 1/16 Filtering: Moderate).



(c) Dense point cloud without masks.
(Resolution: 1/64 Filtering: Moderate).



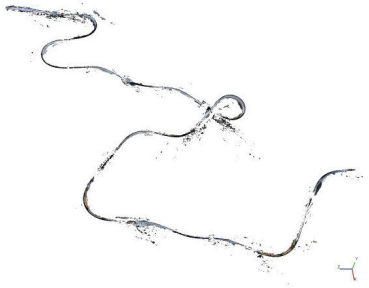
(d) Dense point cloud without masks.
(Resolution: 1/4 Filtering: Aggressive).



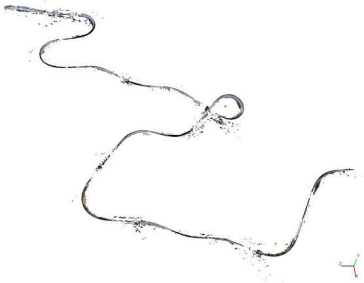
(e) Dense point cloud without masks.
(Resolution: 1/16 Filtering: Aggressive).



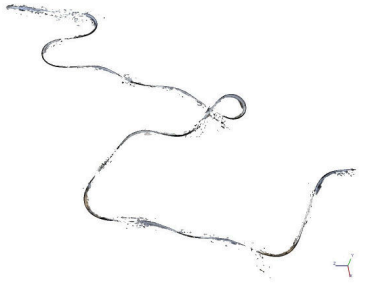
(f) Dense point cloud without masks.
(Resolution: 1/64 Filtering: Aggressive).



(g) Mesh from dense point cloud with masks.
(Resolution: 1/4 Filtering: Aggressive).

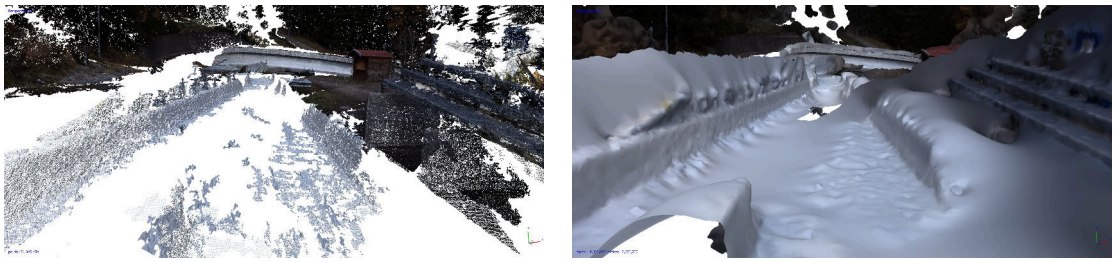


(h) Mesh from dense point cloud with masks.
(Resolution: 1/16 Filtering: Aggressive).

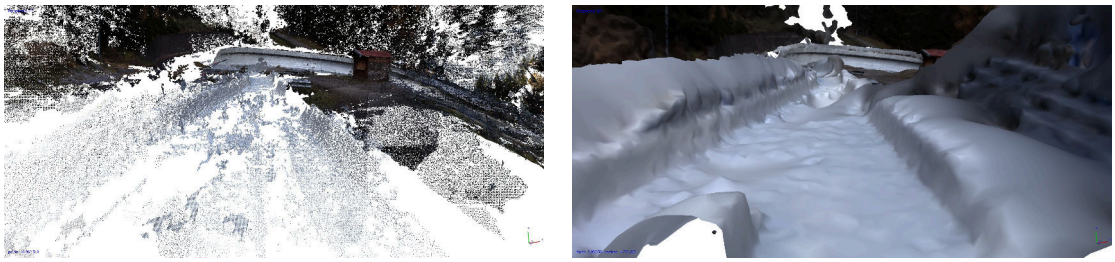


(i) Mesh from dense point cloud with masks.
(Resolution: 1/64 Filtering: Aggressive).

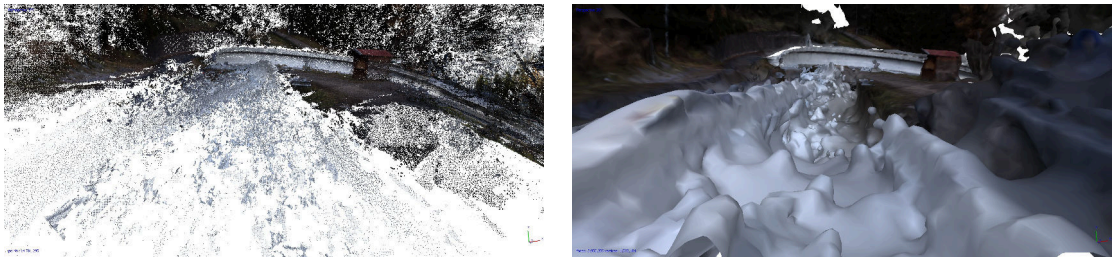
Figure G.5: Point clouds and 3D meshes of the Igls ice track that were generated from data set *IglsIce*26*C*.



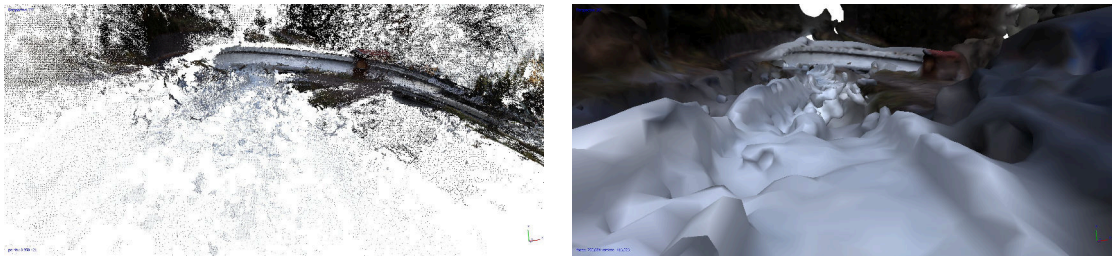
(a) Resolution: 1/16 Filtering: Aggressive.



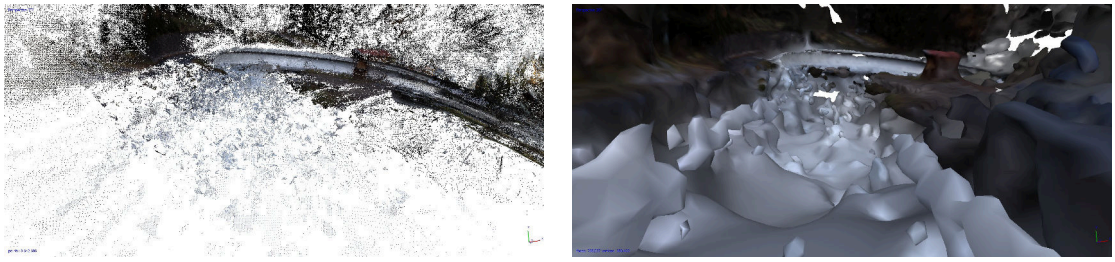
(b) Resolution: 1/64 Filtering: Aggressive.



(c) Resolution: 1/64 Filtering: Moderate.

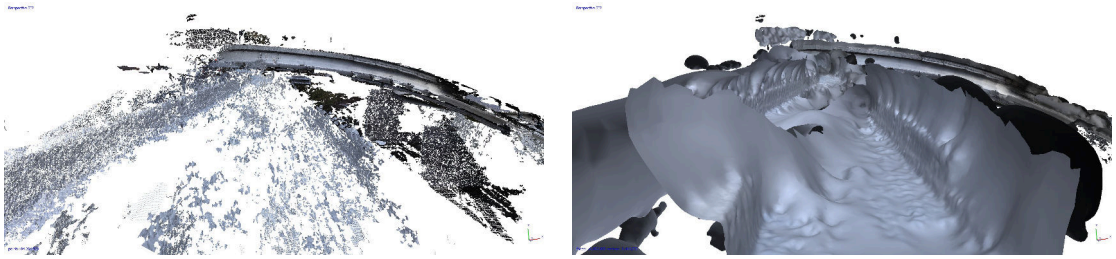


(d) Resolution: 1/256 Filtering: Aggressive.

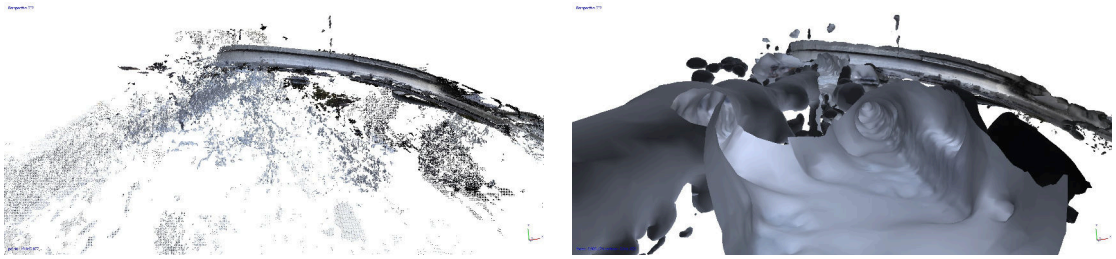


(e) Resolution: 1/256 Filtering: Moderate.

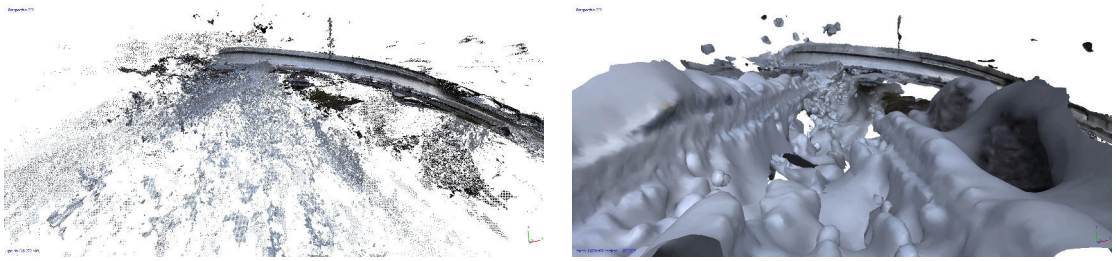
Figure G.6: Dense point clouds, which were generated without masks from data set *IglsIce*26*C*, and corresponding meshes of a segment of the Igls track starting straight.



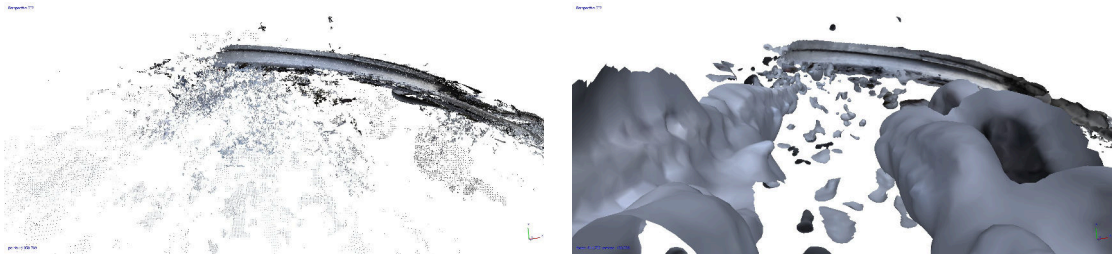
(a) Resolution: 1/16 Filtering: Aggressive.



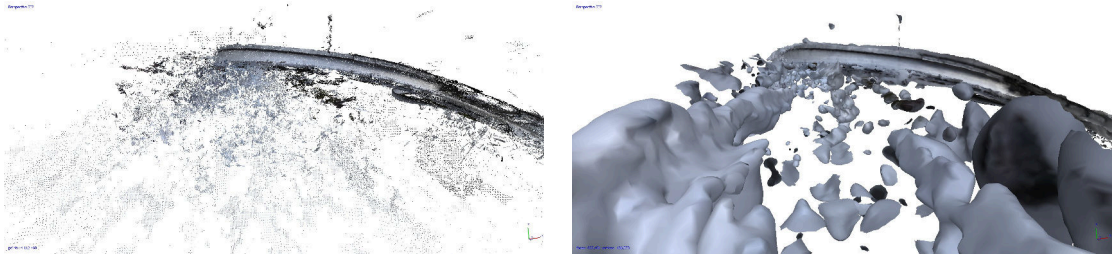
(b) Resolution: 1/64 Filtering: Aggressive.



(c) Resolution: 1/64 Filtering: Moderate.



(d) Resolution: 1/256 Filtering: Aggressive.

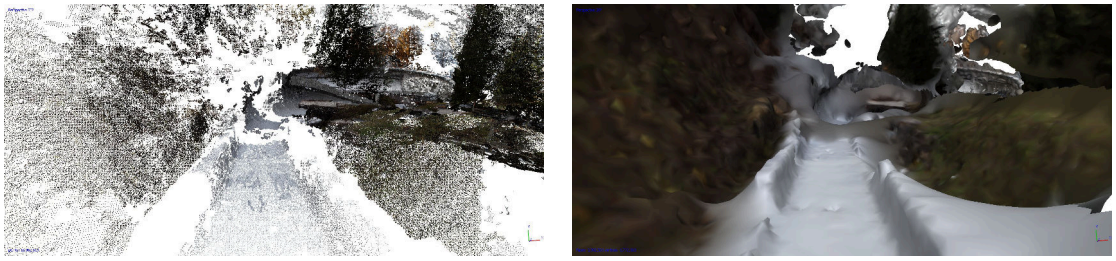


(e) Resolution: 1/256 Filtering: Moderate.

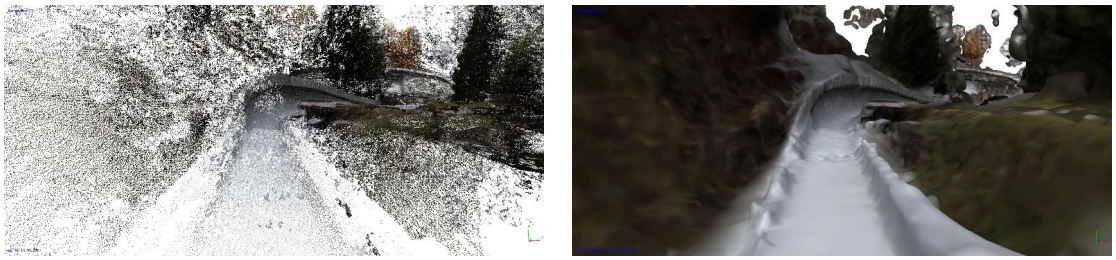
Figure G.7: Dense point clouds, which were generated with masks from data set $IglsIce26C$, and the corresponding meshes of a segment of the Igls track starting straight .



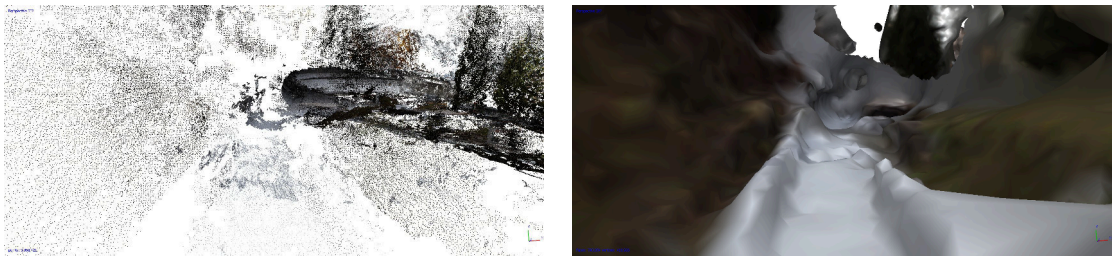
(a) Resolution: 1/16 Filtering: Aggressive.



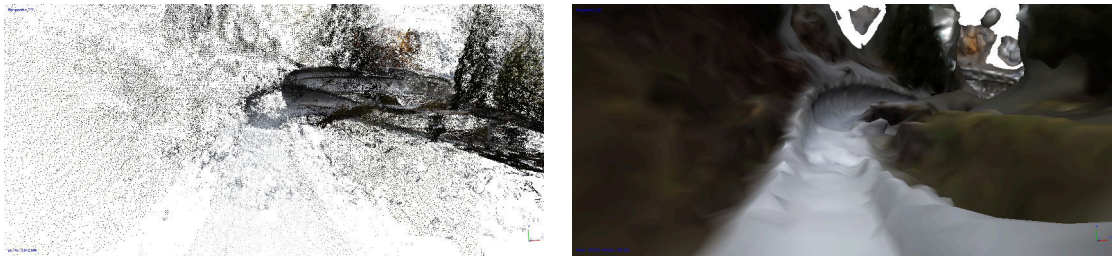
(b) Resolution: 1/64 Filtering: Aggressive.



(c) Resolution: 1/64 Filtering: Moderate.

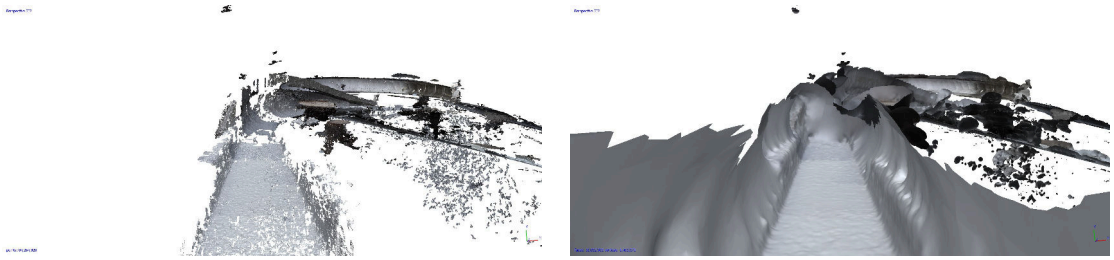


(d) Resolution: 1/256 Filtering: Aggressive.

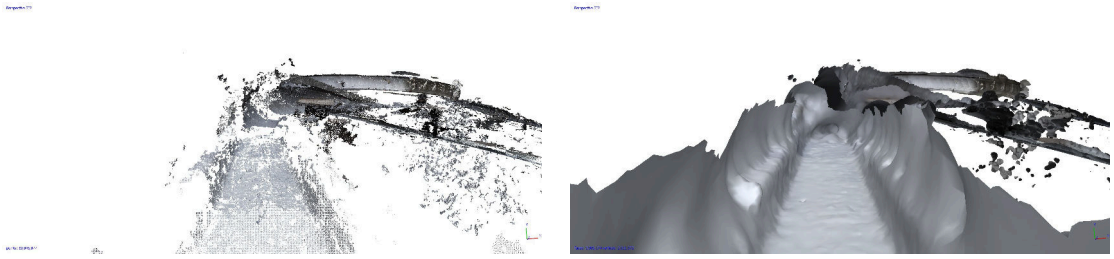


(e) Resolution: 1/256 Filtering: Moderate.

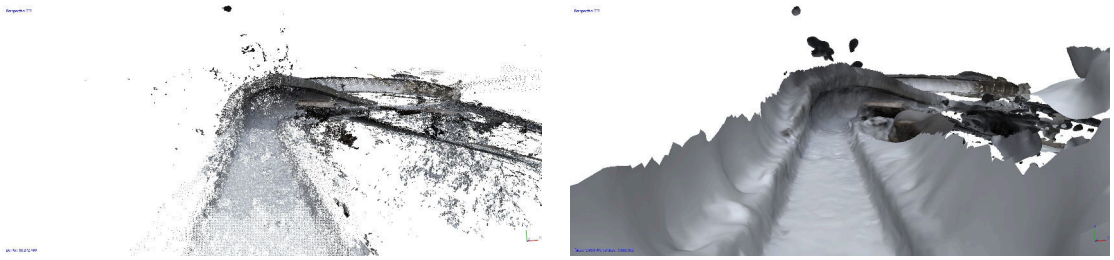
Figure G.8: Dense point clouds, which were generated without masks from data set *IglsIce*26*C*, and the corresponding meshes of a straight segment of the Igls track.



(a) Resolution: 1/16 Filtering: Aggressive.



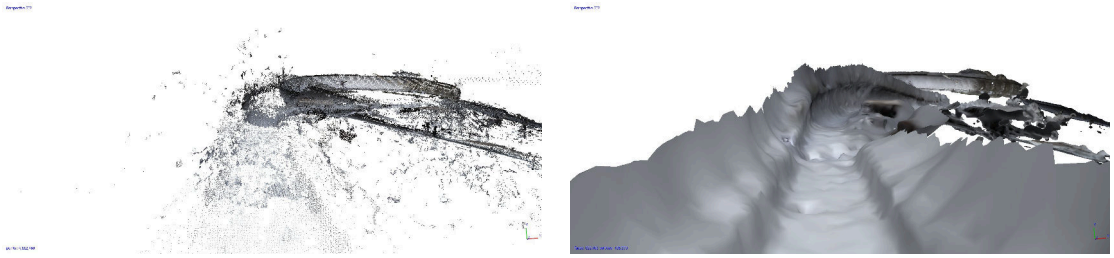
(b) Resolution: 1/64 Filtering: Aggressive.



(c) Resolution: 1/64 Filtering: Moderate.

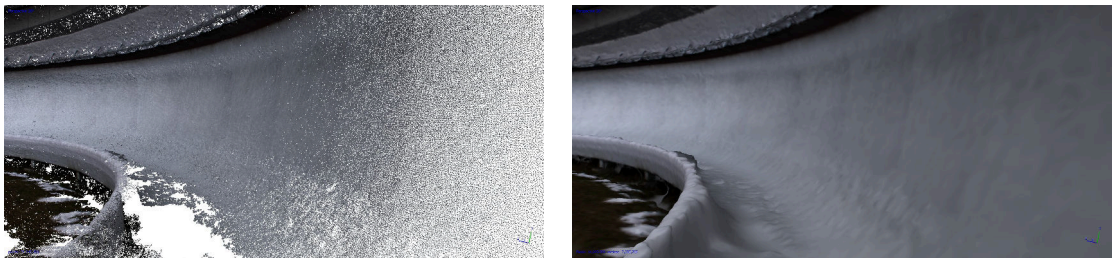


(d) Resolution: 1/256 Filtering: Aggressive.

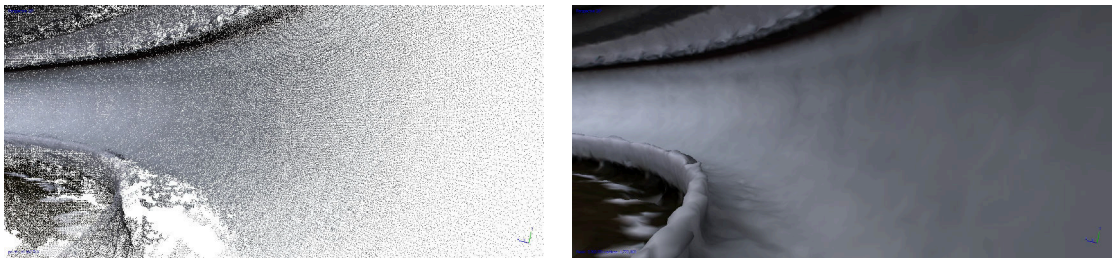


(e) Resolution: 1/256 Filtering: Moderate.

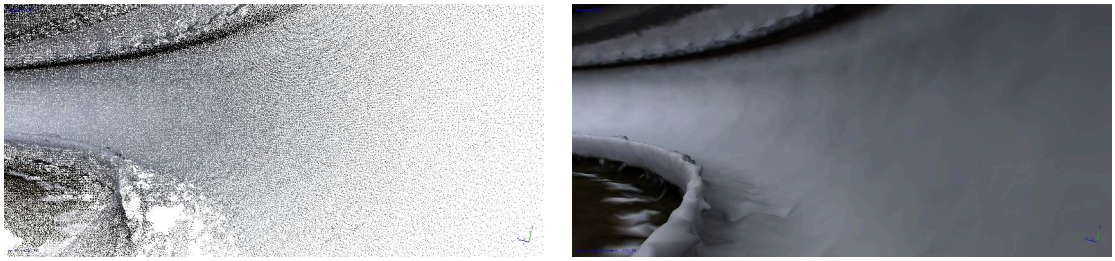
Figure G.9: Dense point clouds which were generated with masks from data set $IglsIce26C$, and the corresponding meshes of a straight segment of the Igls track.



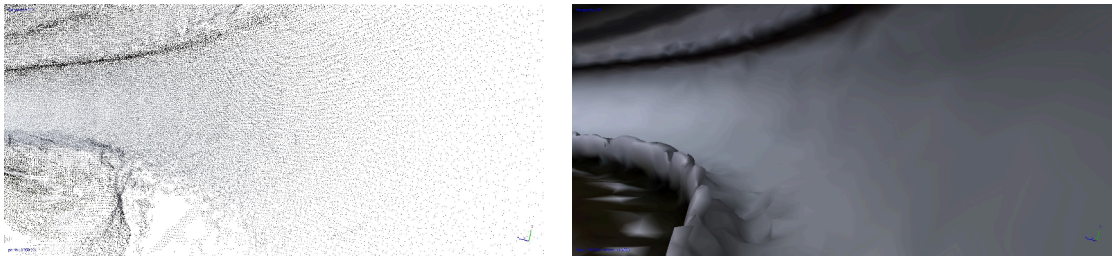
(a) Resolution: 1/16 Filtering: Aggressive.



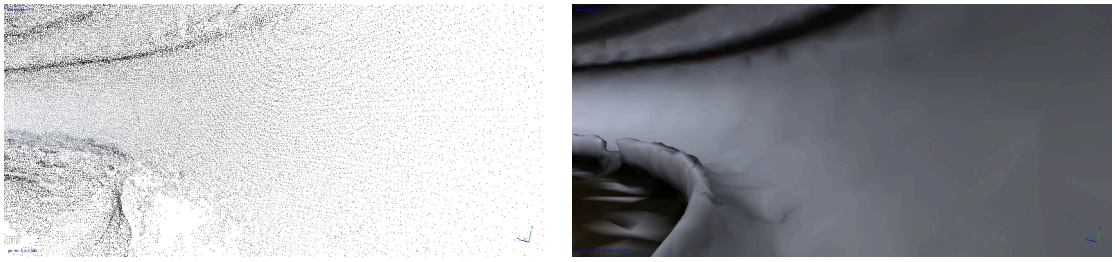
(b) Resolution: 1/64 Filtering: Aggressive.



(c) Resolution: 1/64 Filtering: Moderate.

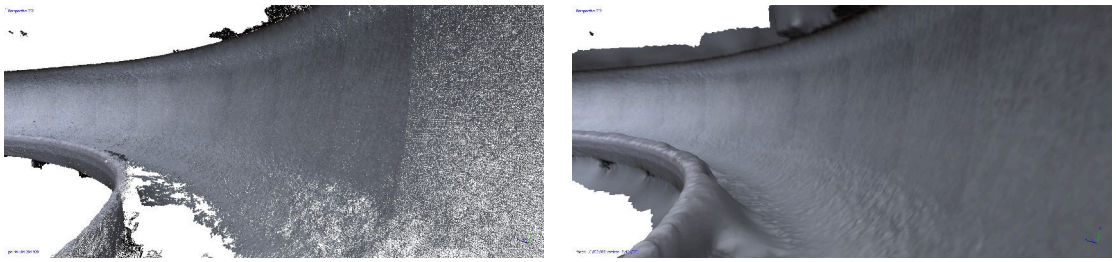


(d) Resolution: 1/256 Filtering: Aggressive.

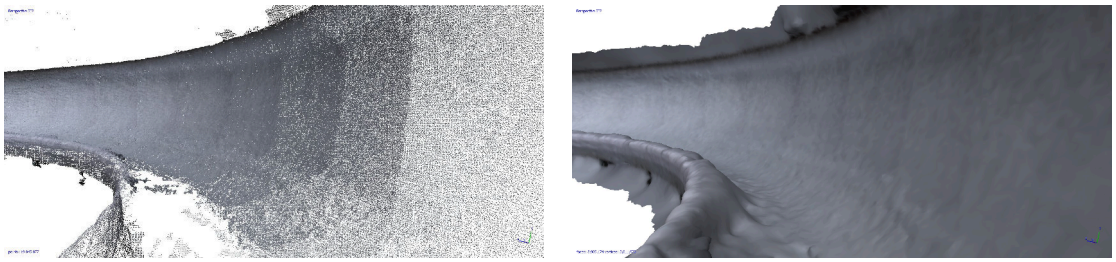


(e) Resolution: 1/256 Filtering: Moderate.

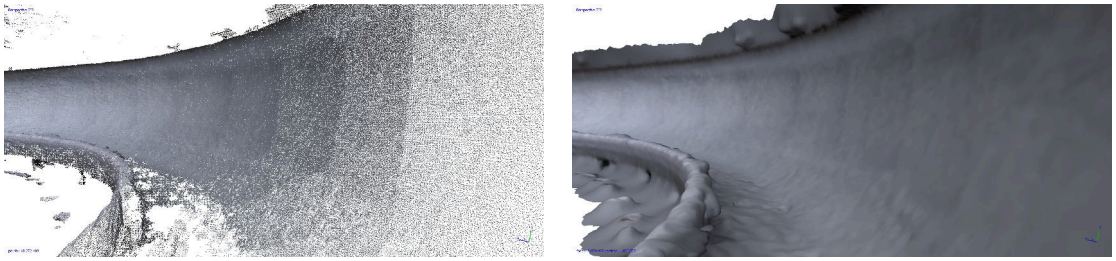
Figure G.10: Dense point clouds, which were generated without masks from data set *IglsIce*26*C*, and the corresponding meshes of a bend segment of the Igls track.



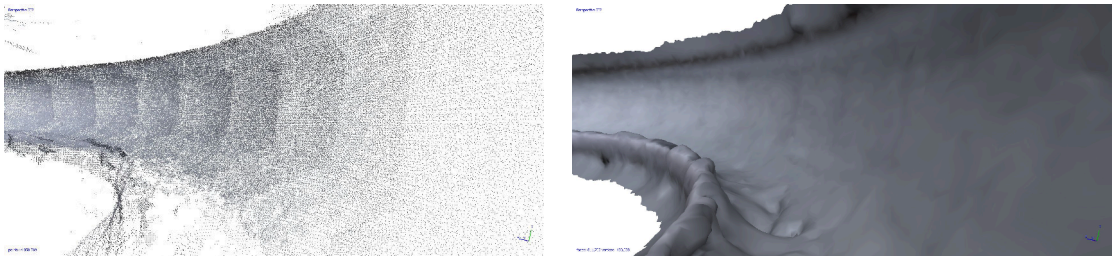
(a) Resolution: 1/16 Filtering: Aggressive.



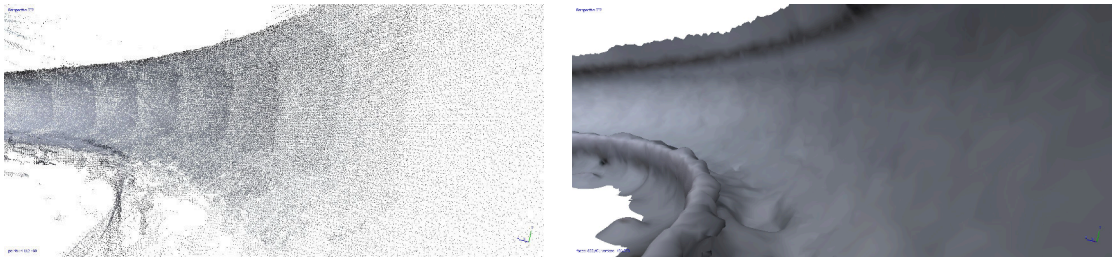
(b) Resolution: 1/64 Filtering: Aggressive.



(c) Resolution: 1/64 Filtering: Moderate.



(d) Resolution: 1/256 Filtering: Aggressive.



(e) Resolution: 1/256 Filtering: Moderate.

Figure G.11: Dense point clouds, which were generated with masks from data set *IglsIce*26*C*, and the corresponding meshes of a bend segment of the Igls track.