EGOMOTION ESTIMATION AND MULTI-RUN DEPTH DATA INTEGRATION FOR 3D RECONSTRUCTION OF STREET SCENES

A DISSERTATION SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Supervisors
Prof. Reinhard Klette
Dr. Chia-Yen Chen

October 2017

By
Hsiang-Jen Chien
School of Engineering, Computer and Mathematical Sciences

Copyright

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Johnny Chin

Signature of candidate

Acknowledgements

I gratefully acknowledge the cast of thousands who supported my journey of discovery. The thesis would not have been done without the unconditional love from my dear mum, dad, grandma, and the rest of my family (including Coffee, an adorable poodle).

I sincerely thank Prof. Reinhard Klette and Dr. Chia-Yen Chen for being admirable captains guiding me through countless challenges on the knowledge pursuing adventure.

I appreciate the support from Reinhard's research group. Mahdi, Junli, Dongwei, and Haokun provided valuable personal advice at the very beginning of my doctoral research. I especially thank Haokun for establishing the trinocular visual odometry framework, which has been increasingly grown into a comprehensive VO/SLAM implementation throughout the progress of my thesis. I thank Noor for being a supportive mate during our 2-month stay at Wuhan, China. I thank Eason Gu, and many of my unnamed friends from a wide variety of countries, for sharing pleasure moments at Auckland. I also thank Dr. Weiqi Yan for providing technical advice for my research.

I thank Dr. Uwe Franke and his team, including Clemens, David, Lukas, Marius, Markus, Nick, Stefan, Simon, and the unnamed rest, for my stay at Stuttgart, Germany. I personally appreciate Nick Schneider for his support as a good workmate and friend as well during my internship at Daimler A.G.

I thank God, for his mercy, grace, wisdom, and words. *"Knowledge puffs up, but love builds up."*

Waiheke Island, September 2017

Hsiang-Jen (Johnny) Chien

Abstract

Digitalization of a 3D scene has been a fundamental yet highly active topic in the field of computer science. The acquisition of detailed 3D information on street sides is essential to many applications such as driver assistance, autonomous driving, or urban planning. Over decades, many techniques including active scanning and passive reconstruction have been developed and applied to achieve this goal. One of the state-of-the-art solutions of passive techniques uses a moving stereo camera to record a video sequence on a street which is later analysed for recovering the scene structure and the sensor's egomotion that together contribute to a 3D scene reconstruction in a consistent coordinate system.

As a single reconstruction may be incomplete, the scene needs to be scanned multiple times, possibly with different types of sensors to fill in the missing data. This thesis studies the egomotion estimation problem in a wider perspective and proposes a framework that unifies multiple alignment models which are generally considered individually by existing methods. Integrated models lead to an energy minimisation-based egomotion estimation algorithm which is applicable to a wider range of sensor configurations including monocular cameras, stereo cameras, or LiDAR-engaged vision systems.

This thesis also studies the integration of 3D street-side models reconstructed from multiple video sequences based on the proposed framework. A keyframe-based sequence bag-of-words matching pipeline is proposed. For integrating depth data

from difference sequences, an alignment is initially found from established cross-sequence landmark-feature observations, based on the aforementioned outlier-aware pose estimation algorithm. The solution is then optimised using an improved bundle adjustment technique. Aligned point clouds are then integrated into a 3D mesh of the scanned street scene.

Publications

Visual odometry based on transitivity error analysis in disparity space - A third-eye approach	H.-J. Chien, H. Geng, and R. Klette. Proc. *Image and Vision Computing New Zealand*, ACM Digital Library, pages 72–77, 2014
Bundle adjustment with implicit structure modelling using a Direct Linear Transform	H.-J. Chien, H. Geng, and R. Klette. Proc. *Computer Analysis of Images and Patterns*, pages 411–422, 2015
Egomotion estimation and reconstruction with Kalman filters and GPS integration	H. Geng, H.-J. Chien, and R. Klette. Proc. *Computer Analysis of Images and Patterns*, pages 399–410, 2015
Multi-frame feature integration for multi-camera visual odometry	H.-J. Chien, H. Geng, C.-Y. Chen, and R. Klette. Proc. *Pacific-Rim Symposium on Image and Video Technology*, pages 27–37, 2015
Multi-run: An approach for filling in missing information of 3D roadside reconstruction	H. Geng, H.-J. Chien, and R. Klette. Proc. *Pacific-Rim Symposium on Image and Video Technology Workshop*, pages 192–205, 2015
When to use what feature? SIFT, SURF, ORB, or A-KAZE features for monocular visual odometry	H.-J. Chien, C.-C. Chuang, C.-Y. Chen, and R. Klette. Proc. *Image and Vision Computing New Zealand*, pages 1–6, 2016
Visual odometry driven online calibration for monocular LiDAR-camera systems	H.-J. Chien, N. Schneider, U. Franke, and R. Klette. Proc. *International Conference on Pattern Recognition*, pages 2848–2853, 2016
Regularised energy model for robust monocular ego-motion estimation	H.-J. Chien, and R. Klette. Proc. *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, volume 6, pages 361–368, 2017
Improved stixel estimation based on transitivity analysis in disparity space	N. H. Saleem, H.-J. Chien, M. Rezaei, and R. Klette. Proc. *Computer Analysis of Images and Patterns*, 2017
Multi-objective visual odometry	H.-J. Chien, J.-J, Lin, T.-K. Ying, and R. Klette. Proc. *Pacific-Rim Symposium on Image and Video Technology*, 2017

Contents

Copyright **2**

Declaration **3**

Acknowledgements **5**

Abstract **6**

Publications **9**

List of Symbols **16**

1 Introduction **18**

1.1 Street scene reconstruction 19
1.2 Background 21
1.2.1 Structure from motion 21
1.2.2 Simultaneous localisation and mapping 22
1.2.3 Visual odometry 23
1.3 Motivations 24
1.4 Contributions 25
1.5 Notation 26
1.6 Thesis organisation 26

2 Literature Review **27**

2.1 Parallel tracking and mapping 27
2.2 Large-scale dense SLAM 28
2.3 SLAM for versatile cameras 29
2.4 Library for visual odometry 30
2.5 Summary 31

3 Scene Structure, Image Correspondence, and Camera Motion **32**

3.1 The triality 32
3.2 Structure recovery 35
3.2.1 2-view triangulation 35
3.2.2 Disparity-depth conversion 42

3.3 Establishing image correspondence 44
3.3.1 Photometric matching 46
3.3.2 Descriptor matching 49
3.4 Camera motion recovery 51
3.4.1 Perspective-n-point (PnP) problem 51
3.4.2 Essential matrix decomposition 54
3.4.3 Correspondence-free methods 56
3.4.4 Iterative energy minimisation 60
3.5 Closure 62

4 Visual Odometry 63
4.1 Overview 63
4.1.1 Feature-based methods 65
4.1.2 Appearance-based methods 67
4.2 A unifying framework 69
4.3 Alignment models 71
4.3.1 Epipolar alignment 72
4.3.2 Projective alignment 74
4.3.3 Photometric alignment 75
4.3.4 Rigid alignment 77
4.4 Egomotion estimation 78
4.5 Post-motion processing 80
4.5.1 Inlier injection 80
4.5.2 Structure recovery and recursive filtering 81
4.6 Experiments 85
4.6.1 Multi-objective egomotion estimation 85
4.6.2 Tracking topology 87
4.7 Closure 89

5 Multiple-run Reconstruction 91
5.1 Overview 91
5.2 Sensor configurations 93
5.2.1 Stereo sequence 93
5.2.2 Monocular-LiDAR sequence 94
5.3 Establishing inter-sequence correspondences 97
5.3.1 Keyframe extraction 98
5.3.2 Bag-of-words model 99
5.3.3 False-positive rejection 102
5.4 Aligning sequences 102
5.4.1 Finding an initial pose 103
5.4.2 Refinement using bundle adjustment 103
5.5 Depth integration and surface reconstruction 108
5.6 Conclusion 111

6 Future Work **112**

A Camera Model and Multiocular Calibration **115**

A.1 Nonlinear camera model . . . 116

A.2 Monocular and stereo calibration . . . 117

A.3 Multiocular calibration . . . 119

A.3.1 Calibration graph and parameters initialisation . . . 119

A.3.2 Nonlinear optimisation . . . 121

A.4 Experiments . . . 123

A.5 Summary . . . 125

Appendices **115**

B Comparative Study on Image Features **129**

B.1 Image features . . . 129

B.1.1 SIFT . . . 129

B.1.2 SURF . . . 130

B.1.3 ORB . . . 130

B.1.4 A-KAZE . . . 131

B.2 Experiments . . . 132

References **138**

Index **147**

List of Tables

3.1 Strategies to solve unknowns under multi-view geometry 34

3.2 Comparison of two basic options for establishing image correspondences. 46

4.1 Summary of alignment model and data terms 79

4.2 Egomotion estimation drifts of different model combinations 86

4.3 Feature tracking topologies. 88

B.1 Comparison of feature extraction 133

List of Figures

3.1 The triality of two-view geometry 33
3.2 Geometry of mid-point triangulation 40
3.3 Error ellipsoid of a point from disparity conversion 44
3.4 Structure reconstruction from a disparity map 45
3.5 Motion recovery from essential matrix decomposition 57
3.6 Motion recovery from direct photometric alignment 59

4.1 Stages of a generic VO implementation 64
4.2 Egomotion estimation of a feature-based VO implementation 67
4.3 Egomotion estimation of a direct VO implementation 68
4.4 Egomotion estimation of a unifying multi-objective VO implementation 70
4.5 RANSAC process to find consensus egomotion among multiple objectives. 71
4.6 Recovery of lost features 82
4.7 Point fusion using recursive Bayesian filter 84
4.8 Drift analysis of VO based on different alignment models 87
4.9 Evaluated motion error among different tracking topologies 90

5.1 Reconstruction of a 120 metres long street-side scene using stereo VO 93
5.2 Street-view of a the reconstructed scene. 93
5.3 Example of a dense depth map from up-sampled LiDAR data 95
5.4 Monocular-LiDAR VO framework results 96
5.5 Reconstruction of a street side using monocular-LiDAR VO 96
5.6 Covisibility matrix 99
5.7 Extracted keyframes based on covisibility 100
5.8 Cross-sequence similarity matrix 101
5.9 Comparison of computation time of a bundle adjustment process. . . 106
5.10 Visualisation of aligned sequences 107
5.11 A closer look at the aligned sequences 107
5.12 Integration of depth data from two runs 110

6.1 A failed case of BoW-based frame-to-frame correspondence establishment 113

A.1 Example of a calibration graph and its MST. 121
A.2 Hessian matrix (in logarithmic scale) of a 3-camera 10-view calibration problem. 124
A.3 Coverage of 38 placements of the calibration target. 125

A.4 Error plot through the global optimisation process. 125
A.5 Reprojection error of control points using pairwise optimised parameters without global bundle adjustment. 126
A.6 Control points reprojected using globally optimised parameters. 126
A.7 Rectified views without global bundle adjustment 127
A.8 Rectified views using globally optimised parameters. 128

B.1 Box plot of evaluated egomotion estimates from different features . . 134
B.2 Inter-frame and accumulated drift errors 135
B.3 Segmented motion error analysis 136

List of Symbols

Linear algebra and set theory

a, b, c	Scalars
$\mathbf{a}, \mathbf{b}, \mathbf{c}$	Vectors
$\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}$	Vectors in homogeneous coordinates
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	Matrices
$\mathbf{A}^\top, \mathbf{A}^{-1}, \mathbf{A}^+$	Transpose, inverse and pseudoinverse of matrix
$\mathbb{R}, \mathbb{R}_+$	Real numbers and positive real numbers
$\sim$	Projective equality
$\leftrightarrow$	Correspondence

Operators

$\|\square\|$	Euclidean norm of a vector
$\| \square \|_{\mathbf{\Sigma}}$	Mahalanobis norm of a vector
$[\square]_\times$	Skew-symmetric form of a vector
π	$\mathbb{R}^3 \to \mathbb{R}^2$ projection function
π^{-1}	$\mathbb{R}^2 \to \mathbb{R}^3$ back-projection function
τ	$\mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^3$ point triangulator
δ	$\mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}_+$ distance function
$\circ$	Pose concatenation

Special matrices

$\mathbf{R}$	Rotation matrix in $\mathbb{SO}(3)$
$\mathbf{T}$	Euclidean transformation in $\mathbb{SE}(3)$
$\mathbf{K}$	Camera matrix
$\mathbf{P}$	Projection matrix $\mathbf{P} = \mathbf{K}\,[\mathbf{R}\;\;\mathbf{t}]$
$\mathbf{E}$	Essential matrix $\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R}$
$\mathbf{F}$	Fundamental matrix $\mathbf{F} = \mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{\top}$
$\mathbf{\Sigma}$	Variance-covariance matrix
$\mathbf{J}$	Jacobian matrix
$\mathbf{H}$	Hessian matrix

Geometry and image

$\mathbf{x} = (x, y)^{\top}$	Pixel coordinates in observed image plane
$\mathring{\mathbf{x}} = (\mathring{x}, \mathring{y}, 1)^{\top}$	Normalised pixel coordinates in canonical image plane
$\mathbf{y} = (X, Y, Z)^{\top}$	World coordinates in $\mathbb{R}^3$
$\mathbf{t}$	Translation vector in $\mathbb{R}^3$, as a column vector
ξ	Twist in Lie-algebra $\mathfrak{se}(3)$, minimally representing a Euclidean transformation
i	Index of a tracked landmark
j	Index of a frame
k	Index of a camera; iteration number
I, Ω	Image and its domain
χ	Image feature
$\mathcal{F}$	Set of image features
$\mathcal{M}$	Mapping of features
Φ, φ	Objective and error residual functions

Chapter 1

Introduction

The reconstruction of 3-dimensional (3D) environments has been actively studied since the early development of modern computer science. In the last two decades, a number of breakthroughs have been made in a variety of fields including computer vision, photogrammetry, robotics, and optics. Moreover, the computational capability of machines nowadays is being rapidly boosted at an exponential scale. These achievements, altogether, bring into the reality the creation of high-definition digital models of 3D scenes at a large scale.

This chapter first introduces in Section 1.1 existing methods for street scene reconstruction, among which the computer vision-based approach is selected for further investigation. In Section 1.2, the background of related academic fields is briefed. We then identify in Section 1.3 gaps existing in the fields, and discuss in Section 1.4 how they can be filled-in by contributions made by this research. Section 1.5 defines mathematical notions used throughout the thesis. Section 1.6 outlines the structure of this thesis.

1.1 Street scene reconstruction

Accurate recovery of depth information of real-world scenes has been a fundamental, yet highly active research topic in the field of computer vision. The acquisition of reliable 3D information on street sides plays an important role in many advanced technologies such as driver-assistance systems (Morris et al., 2009), collision avoidance (Nedevschi, Bota & Tomiuc, 2009), autonomous vehicles (Thrun, 2010), urban planning (Hu, You & Neumann, 2003), or geosciences (Westoby, Brasington, Glasser, Hambrey & Reynolds, 2012).

A common technique to build 3D models of an urban scene is based on *light detection and ranging* (LiDAR) technology. To collect scans of a scene, a laser scanner is set up, either at a stationary point or a moving vehicle on the ground [i.e. *terrestrial laser scanning* (TLS)], or mounted on an flying aircraft [i.e. *airborne laser scanning* (ALS)]. The former setup provides highly accurate reconstruction of building facades and street objects within a few blocks, while the latter one builds urban *digital elevation models* (DEM) in a wider region.

Despite the accuracy and effectiveness of laser scanning technology, LiDAR still comes with a high cost, is not recording surface colour, and provides limited information for tracking visible information from frame to frame. Image-based 3D reconstruction offers an affordable alternative to approach these problems. The stereo-vision technique, for example, recovers a dense scene structure using a conjugate image pair captured by two calibrated cameras, also possibly at a high frequency (e.g. 30 Hz). To extend the scanning range, well-calibrated image sensors are mounted on a vehicle and rigidly moved, allowing a continuous collection of depth data in the scene. As the *ego-vehicle* (i.e. the vehicle where the system is installed) moves, the sensor's motion has to be recovered, such that the 3D points, measured in different time frames, can be converted into a consistent coordinate system.

Occlusion of objects is a major problem in 3D reconstruction. Objects partially occluded in a single run lead to incomplete reconstructions with missing depth information. To address this problem, the scene must be scanned from multiple viewing directions, and a sophisticated algorithm is required to merge multi-run scans such that a complete 3D reconstruction can be built (Rusu, Blodow, Marton & Beetz, 2008; Zeng & Klette, 2013).

The adopted egomotion[1] estimation strategy is crucial to the process of merging 3D data collected in different runs. In a single run, an inaccurate egomotion estimator may cause significant drifts between frames, making the alignment of points far from being consistent. As a result, the transformed point cloud is distorted, and, in turn, imposes a difficulty in registering multiple-run point clouds.

To ensure the validity of recovered egomotion, the estimator must track and use temporally consistent feature points. In a typical urban traffic scene, however, this can be challenging due to the complexity and dynamics of the street sides. Identifying moving objects (i.e. vehicles, pedestrians, and so forth) is a necessary step in such a scenario. Also, the complexity of a street-side scene may result in a large number of image feature points. An efficient and robust temporal feature-tracking mechanism is therefore desired.

Furthermore, to improve the coverage and accuracy of the reconstructed scene, it may also be scanned using different types of sensors. For example, results obtained from ground-level scans with a vehicle-mounted stereo vision system can be merged with a top-view reconstruction of the street from a monocular camera that is attached to a flying drone.

The reported research aims at achieving comprehensive street-side reconstruction through developing a multi-sensor multi-scan merging methodology, and an improved

[1] Egomotion refers to the motion of a moving sensor. Latin *ego*, first person singular personal pronoun, is chosen by the translator of Sigmund Freud's writings to translate the use of German *Ich* as a noun to English. Literally, egomotion therefore means *self motion*.

egomotion estimation strategy for achieving the required accuracy in camera-motion recovery.

1.2 Background

This section walks through three topics that are closely related to this research, from a historical and a technical perspective.

1.2.1 Structure from motion

Structure recovery from images, taken at different poses[2] is known as *structure from motion* (SfM). It has been well-studied in the field of photogrammetry, since its wide application in remote sensing and geosciences in the early 1990s (Westoby et al., 2012). Thanks to the advances in the theories of algebraic factorisation and auto-calibration (Longuet-Higgins, 1981; Luong & Faugeras, 1997; Kanade & Morris, 1998), the application of SfM has been extended from computer scientists to general users.

Nowadays, the SfM technology is available to the public; one can easily reconstruct 3D structure of a scene in a few clicks, using either a commercial or an open source implementation (Snavely, 2008; C. Wu, 2011). Remarkably, it has recently shown the possibility for automatic 3D model reconstruction from photos on the internet (Snavely, Seitz & Szeliski, 2008).

A general SfM problem aims to estimate the structure of a stationary scene from many of its projective measurements - images, taken by one or many uncalibrated cameras in arbitrary many poses. Solving such an inverse problem is challenging, as the solutions lie in a high-dimensional space. An instance of a 10-view reconstruction problem can, for example, lead to a non-linear solution space above one thousand dimensions.

[2] A *pose* is defined by position and direction.

Historically, the SfM techniques follow a multi-stage pipeline including feature extraction, correspondence establishment, camera parameter recovery and finally structure recovery. In general, parameters are locally estimated using a fast linear solver, followed by a global non-linear optimisation stage known as *bundle adjustment* (BA). Due to computational complexity, an SfM problem is usually solved offline.

1.2.2 Simultaneous localisation and mapping

In robotics it is often required to locate an agent and the 3D structure of its surrounding environment. A technique, jointly estimating the location and the scene structure, is known as *simultaneous localisation and mapping* (SLAM) (Cadena et al., 2016). The SLAM problem can be approached using different types of sensors, such as laser range-finders (i.e. LiDAR), radar, sonar, and optical sensors.

In the case that the estimation is based on image data, it is known as *visual SLAM* (V-SLAM) (Karlsson et al., 2005), which can be understood as a specialised SfM problem. Unlike a general SfM solver, a SLAM [3] system performs pose estimation and structure computation on-line, from a temporal continuous image sequence in an incremental manner. The sensors are also assumed to be well-calibrated. As the estimation has to be done on-line, critically in real-time, a SLAM algorithm needs to take trade-offs between accuracy and computation efficiency.

When a global positioning sensor is not available, the localisation has to be actualised by integrating differential pose measurements over time. The technique is called *dead reckoning* [also known as *deduced reckoning* (DR)], in navigation (Gehrig & Stein, 1999).

The differential pose can be derived from, for example, an *inertial measure unit* (IMU) or from images. The latter case is known as *visual odometry*.

[3] By "SLAM" it specifically refers to "V-SLAM" through this thesis, unless explicitly specified otherwise.

A typical SLAM system has the ability to recognise a visited place. Such an ability allows the system the removal of an accumulated positioning error by enforcing consistency among multiple measurements on repeatedly observed scene structures, using bundle adjustment. Such a drift-suppression strategy is known as *loop closure*. In addition to loop closure, a modern SLAM system also performs incremental frame-by-frame integrations on the estimated parameters by means of a recursive filter, such as the *extended Kalman filter* (EKF) (Durrant-Whyte & Bailey, 2006).

1.2.3 Visual odometry

The term visual odometry (VO) was coined in 2004, as an analogue to wheel odometry (Nistér, Naroditsky & Bergen, 2004). It provides a way to estimate the trajectory of a moving camera from an image sequence. VO techniques are often adopted to complement a global positioning system (GPS) under circumstances that satellite signals are blocked (e.g. when a vehicle is driving in a tunnel). Following VO successes with Mars exploration rovers (Maimone, Cheng & Matthies, 2007), VO has been widely adopted by many applications such as SLAM, intelligent vehicles, indoor navigation, or augmented reality (AR) (Klein & Murray, 2007; Scaramuzza & Fraundorfer, 2011).

The core of VO is to solve a 2-view camera-pose recovery problem in a special form, known as *egomotion estimation*, where the first and second views are taken by the same camera (and the relative pose is therefore the camera's egomotion). The solution is dependent on the type of sensors. Early works focused on the use of stereo cameras, leading to solving a 3D-to-3D or 3D-to-2D registration problem. In recent years, much effort has been done in monocular VO, which is often considered to be a more challenging case as an accurate estimate of scene structure is not directly available from monocular data.

The development of VO has led to a separation into two different paths, namely *appearance-based* or *feature-based* techniques. Feature-based approaches follow feature-matching and tracking pipelines developed in the area of SfM. These approaches dominated the development of VO for decades. Appearance-based VO, on the other hand, makes direct use of image intensities, thus also known as *direct methods*. Early direct methods are influenced by optical-flow estimation and SfM techniques as developed in the photogrammetry community (Irani & Anandan, 1999). After decades of oblivion, the direct methods have quickly become popular in the last few years, thanks to the advance of GPU computing (Engel, Sturm & Cremers, 2013). In 2007, the first symbolic implementation of sparse direct VO is presented in the context of augmented reality (Klein & Murray, 2007).

1.3 Motivations

Both SfM and SLAM techniques provide the possibility to achieve 3D street scene reconstruction. However, we[4] found gaps among these techniques when it comes to multiple-run integration. The SfM pipeline considers unordered images taken in arbitrary positions and offers a many-to-many cross-matching functionality to sort out the informational correlation between view points, which is also useful to align sequences taken in multiple runs. It is, however, very computational expensive as a sequence of a street, considered in this research, may contain thousands of images. Furthermore, the images are assumed to be collected by some well-calibrated sensors in temporal order. Taking these constraints into account saves, indeed, unnecessary computational requirements by avoiding cross-frame all-to-all matching as well as optimising well-calibrated sensor parameters.

On the other hand, methods from the SLAM family usually deploy efficient strategies

[4] The use of "we" throughout this thesis is purposeful. It is used to involve the reader with the thesis as recommended by Knuth (Knuth, Larrabee & Roberts, 1989).

to recognise a visited place and allow data, collected at different times, to be fused. However, most of the SLAM techniques are designed to perform on-line map integration only, which might sacrifice accuracy in 3D reconstruction in order to maintain a moderate frame rate. Also, most of the techniques found in the literature consider only specific sensor models. Difficulties arise when one is trying to apply a stereo-based SLAM algorithm to an airborne monocular vision system.

It is hard to find a versatile framework that comes with a good flexibility to support the extension into a combination of different optical or range sensors such as monocular camera, stereo vision, multi-ocular vision, and LiDAR (Moosmann & Stiller, 2011). Such drawbacks motivated our research to develop an appropriate framework for *multiple-sequence mapping*.

1.4 Contributions

The objective of this research is to design, implement, and evaluate a novel reconstruction framework that can be applied to build 3D representation of a sequence and merge reconstructions built from multiple sequences, following a versatile design that should be applicable to monocular, stereo, and multi-ocular optical sensors, optionally also equipped with a LiDAR.

To this goal, we propose a unifying visual odometry framework which is based on a universal pipeline, to achieve accurate egomotion estimation for a variety of sensor combinations. The framework treats the egomotion estimation problem as an abstract *nonlinear energy minimisation process*, and integrates multiple criteria commonly adopted in the literature to build a robust objective function. The fusion of components from different mathematical models is formulated in a Bayesian framework, based on recursive filtering techniques commonly adopted by a real-time SLAM implementation.

The work also fills in the gap between SfM and SLAM techniques by transforming an incremental on-line pose estimation process in the SLAM category into offline frame-registration and multiple-view structure refinement processes. The registration is initiated by using a bag-of-visual-word technique to discover the correspondences between key frames. Structures and poses of two matched sequences are then rigidly aligned using large-scale offline bundle adjustment.

1.5 Notation

Throughout this thesis, we use fonts to differentiate mathematical entities of different types. A scalar is written in italic font, a vector is in bold lowercase, and a matrix in bold uppercase. For example, in $\mathbf{Ax} = \lambda\mathbf{x}$, $\mathbf{A}$ is a matrix, $\mathbf{x}$ is one of its eigenvectors, and λ is the corresponding eigenvalue. The dimensionality of an entity is written in uppercase, such as $N_{\text{ROW}} \times N_{\text{COL}}$.

We use subscript to denote a subelement of an entity; $\mathbf{a}_i$ refers to the i-th row of $\mathbf{A}$, and a_{ij} is an entry of $\mathbf{A}$. By default, vectors are in column-vector form, a matrix is applied to a vector by right multiplication, and matrices are concatenated in pre-multiplication order. For example, applying a sequence $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$ of linear transformations to a vector $\mathbf{x}$ yields $\mathbf{y} = \mathbf{A}_3 (\mathbf{A}_2 (\mathbf{A}_1 \mathbf{x})) = \mathbf{A}_3\mathbf{A}_2\mathbf{A}_1\mathbf{x}$.

1.6 Thesis organisation

The thesis is organised as follows. Chapter 2 provides a walk-through for state-of-the-art approaches. Chapter 3 describes the theoretical foundation of this research that exist in the area; based on this a novel visual odometry framework is formulated in Chapter 4. This framework is then used to build 3D street-side reconstructions from multiple sequences, as detailed in Chapter 5. Cahpters 4 and 5 define the author's contribution to the field. Chapter 6 discusses possibilities to further extend this research.

Chapter 2

Literature Review

In this chapter four representative implementations in the context of egomotion estimation and 3D mapping are reviewed. These seminal work have demonstrated remarkable solutions to the SLAM/VO problems in terms of novelty and effectiveness, hence considered influential milestones in the field.

The VO and SLAM problems have been studied for decades. In the literature these two loosely related topics are often approached together. A clear boundary divides the existing works into two categories, namely the feature-based methods and the direct methods. The former solves the egomotion estimation by matching a sparse set of feature points, while the latter relies on the patch-based search directly using dense image intensities. In the following sections two of highly cited researches are reported for each category.

2.1 Parallel tracking and mapping

Klein el. al. formulated the SLAM problem as two separate parts, namely tracking and mapping (Klein & Murray, 2007), in their work on *Parallel Tracking and Mapping*

(PTAM). The modularized design allows tasks in a monocular SLAM system to be efficiently parallelised. The incremental mapping methods by the time solve the egomotion estimation and structure update together at every frame, while Klein et. al.'s work is one of the early attempts that decouples the process for real-time applications such as AR with a monocular mobile camera.

Following a coarse-to-fine scheme, the tracking starts with the search of sparse correspondences for a small set of features from accelerated segment test (FAST) (Rosten & Drummond, 2005) in the down-sampled images. An egomotion hypothesis is estimated based on the coarse correspondences and used in turn to establish the image correspondences for a larger feature set in the images at higher resolution. All the tracked image features are then utilised to yield the final pose estimation. A pose hypothesis driven affine warping is applied to increase the accuracy of patch matching and the range is bounded to avoid an exhaustive search in the whole image.

In the mapping module the 3D structure of sparse scene points is continuously maintained based on the selected keyframes. The map is initialised from a dense depth map, which is obtained from a user-guided left-right temporal stereo image pair, followed by an essential matrix decomposition and a two-view triangulation that will be introduced in Chapter 3. As more keyframes are inserted, the map is augmented by finding new observations of the existing features along epipolar lines. PTAM also deploys bundle adjustment (BA) in both local and global scales in each module.

2.2 Large-scale dense SLAM

A recent monocular SLAM approach known as LSD-SLAM has attracted significant attention (Engel, Schöps & Cremers, 2014). The proposed direct SLAM follows the keyframe-based separation of tracking and mapping tasks as suggested by PTAM. The tracking is, however, done on a dense subset of pixels (hence *semi-dense*) instead of sparse image features. The pose of the camera with respect to the active keyframe and point correspondences are solved simultaneously by photoconsistency maximisation (explained in Section 3.3.1).

In the mapping part a dense depth map is continuously updated as a new frame arrives. The update also takes a dense error map into account to actualise a recursive filtering. When a key frame is detected, the updated depth map is locked and used together with the last keyframe's depth map to estimate the egomotion of the camera, by means of a nonlinear minimisation of a variance-normalised energy model. The work also proposes a data-driven normalisation technique to suppress the scale drift problem that encountered in monocular mapping.

Due to the random initialisation at the beginning of LSD-SLAM, the tracking is not guaranteed to converge to a valid configuration. However, such issue can be easily solved using a more robust bootstrapping technique (for example, the left-right temporal stereo as used by PTAM).

2.3 SLAM for versatile cameras

In the feature matching-based SLAM category, the ORB-SLAM based on the oriented FAST and rotated binary robust independent elementary (ORB) features, represents one of the state-of-the-art implementations (Mur-Artal, Montiel & Juan Tardós, 2015). The revised design of ORB-SLAM is composed of three parallel threads respectively working on tracking, mapping and loop closing. The SLAM system finds, extracts and matches ORB features (see Appendix B.1.3 for details) and utilises either a stereo camera pair or a RGB-D sensor to solve for the camera's egomotion. In particular, the egomotion estimation is done by conducting a local bundle adjustment that minimises *reprojection error* (RPE) defined by the projective alignment criterion. The details of the process will be discussed later in Chapter 4.

Once the egomotion is solved, the landmarks in a local map that consists of last few keyframes are projected into the current frame and searched for missing correspondences. The search range is bounded to a small region and based on descriptor matching.

The ORB-SLAM also classifies the tracked keypoints according to their depths to further improve numerical stability in triangulation. The far points are only triangulated when being observed in multiple views. The number of close points is constantly tracked as an indicator for new keyframe insertion.

The descriptors of the extracted ORB features are also used for place recognition. When a previously visited scene is detected, loop closure is enforced by performing a full BA.

2.4 Library for visual odometry

A state-of-the-art software library for stereo VO known as LIBVISO was published by Geiger et. al. in 2011. Similar to ORB-SLAM, the implementation also follows a descriptor-based feature tracking pipeline. The blob and corner features are identified from the images, and the matching of features is based on similarity in the responses of the feature-centred patches to a Sobel filter. The tracking of features is done in a robust manner that covers the stereo images in two consecutive frames, forming a four-camera circular matching topology. A feature is tracked starting from left image of current frame to the right image, then through to the right image of last frame back to the left image of last frame, and finally back to the left image of current frame. If a feature is not mapped to its originating location, the tracking is considered invalid and will be excluded from the egomotion estimation stage.

The speed of matching has been improved by using a small subset of image features to provide statistics hints that are useful to bound the search space. From the established image correspondences, the egomotion is solved by the minimisation of RPE with respect to the left and right cameras. The process is slightly different from the ORB-SLAM's local bundle adjustment by an additional *random sampling consensus* (RANSAC) technique that selects only a subset of image correspondences to participate the adjustment, which is later verified by the full set. The egomotion refined by all the inliers will be eventually given to a Kalman filter to yield the final estimate assuming a constant acceleration motion model.

Despite the library focuses on the VO algorithms, it also comes with a limited mapping ability to integrate multiple depth maps into a filtered reconstruction. The fusion is done by projecting all the 3D points to a selected reference frame and average the depths mapped to the same pixel. Although the naive approach is prone to occlusion, it provides a fast approximation of multi-view reconstruction of scene structure.

2.5 Summary

The reviewed work are limited to a specific sensor configurations. For example, LIBVISO is originally designed for a stereo vision and LSD-SLAM is for monocular sequence. Furthermore, these work solve the egomotion estimation based on a single model (photoconsistency, reprojection error, etc).

Chapter 3

Scene Structure, Image Correspondence, and Camera Motion

This chapter covers the underlying mathematical models among three related concepts in multi-view geometry, based on which the SLAM and VO problems are formulated.

3.1 The triality

Fundamentals of many problems in computer vision and photogrammetry relate to studies between three interconnected entities, namely 3D structure, camera pose and image correspondence. Camera calibration, for example, uses structure-image correspondences to estimate projection parameters and camera poses (see Appendix A for details). A calibrated multi-view vision system, on the other hand, reconstructs 3D structure from established image correspondences.

In the context of SLAM, a vision system (the *agent*) is moving around a scene, makes observations about the scene's structure, and uses correspondences of tracked scene points to deduce the agent's location.

The *scene structure* refers to the 3D geometry of static, rigid, and salient objects in the scene (i.e. this is not a mathematically precise definition). As this structure does not change while the sensor is moving, the only factor affecting the observations would be the *motion* of the agent. Therefore, this poses an inverse problem of recovering the motion from observed changes in scene points.

Let $\mathbf{x} = (x, y)^\intercal$ and $\mathbf{x}' = (x', y')^\intercal$ be the observations of a 3D point $\mathbf{y} = (X, Y, Z)^\intercal$ via projection functions π and π', respectively. Let $\mathbf{R} \in \mathbb{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ be the rotation matrix and translation vector, respectively, together representing the Euclidean transform caused by the motion from a first to a second view. Formally, this is expressed by

$$\begin{aligned} \mathbf{x} &= \pi(\mathbf{y}) \\ \mathbf{x}' &= \pi'(\mathbf{R}\mathbf{y} + \mathbf{t})\,. \end{aligned} \tag{3.1}$$

The described geometrical configuration establishes a *triality* among scene structure, camera motion, and image correspondences (i.e. given a sufficient number of known entities in any two of the three spaces, missing entities in the third space can be uniquely recovered.

The geometry of these mappings π and π' is addressed in Fig. 3.1. Scene structure,

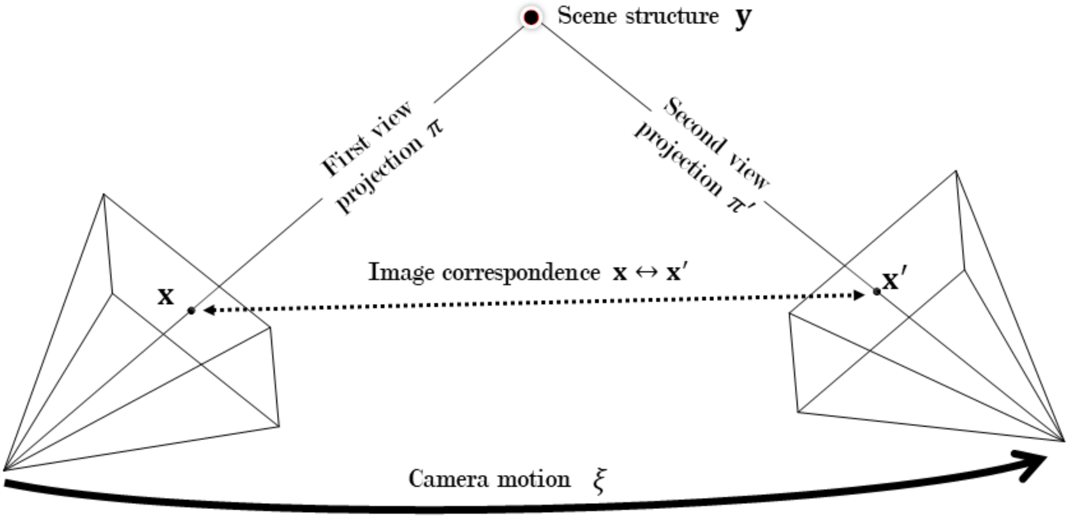

Scene structure $\mathbf{y}$
First view projection π
Second view projection π'
$\mathbf{x}$
Image correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$
$\mathbf{x}'$
Camera motion ξ

Figure 3.1: The triality of two-view geometry

Table 3.1: Strategies to solve unknowns under multi-view geometry

	Scene structure	**Image correspondence**	**Camera motion**	**Strategy**
0	Known	Known	Known	Trivial case.
1	Unknown	Known	Known	Perform back-projection to triangulate scene structure. Camera motion between at least two views is required; extensible to multi-view triangulation (Sec. 3.2).
2	Known	Unknown	Known	Perform forward projection. When the uncertainty of both the estimated scene structure and camera motion is high, it is preferable to perform feature matching (Sec. 3.3.2). With a good confidence in motion estimation, bounded search along epipolar lines is more efficient (Sec. 3.3.1).
3	Known	Known	Unknown	Solve *perspective-n-point* (PnP) problem to recover camera motion. Minimally, three correspondences are required to uniquely determine the motion (Sec. 3.4.1).
4	Unknown	Unknown	Known	Perform unbounded 1D search along epipolar lines (Sec. 3.3.1), then do triangulation (see Case 1).
5	Unknown	Known	Unknown	Perform an essential matrix decomposition to recover camera motion (Sec. 3.4.2) followed by triangulation (see Case 1). The absolute scale of translation remains unknown, so does the scale of the triangulated structure.
6	Known	Unknown	Unknown	Perform feature matching to establish image correspondences then solve a PnP problem (see Case 2). Alternatively one may perform correspondence-free pose estimation (Sec. 3.4.3).
7	Unknown	Unknown	Unknown	Establish image correspondences (see Case 6), perform essential matrix decomposition then do triangulation (see Case 5).

camera motion, and image correspondences define three categories of linked entities in the context of multi-view geometry. Due to camera motion ξ, a 3D point $\mathbf{y}$ is observed at two different image locations $\mathbf{x}$ and $\mathbf{x}'$. This linkage creates the mentioned triality, given a sufficient amount of constraints. Note that the camera motion is represented by the Lie-algebra entity ξ that minimally defines a 3D Euclidean transform. Such a representation is detailed later at the end of Section 3.4.

Table 3.1 summarises a comprehensive list of strategies for solving problems in SLAM and VO.

3.2 Structure recovery

Recovery of structure from imagery data is an essential building block of a visual mapper. This section reviews the structure recovery problem in its simplest form, namely 2-view point triangulation. A special case, that is frequently used for retrieving dense reconstructions from two rectified views, is then given in Section 3.2.2. This case is then extended for multiple views in Section 3.3.1.

3.2.1 2-view triangulation

A 2-view point triangulation function $\tau : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^3$ finds the 3D coordinates $\mathbf{y} = (X, Y, Z)^\intercal$ of a scene point, given its corresponding projections $\mathbf{x} \leftrightarrow \mathbf{x}'$ in two views. We assume that the pose of the second view with respect to the first one, specified by $(\mathbf{R}, \mathbf{t})$, and the projection functions π, π' are known to allow intersecting the back-projected rays.

The process of τ is often over-determined as there are three unknowns in $\mathbf{y}$ while $\mathbf{x}$ and $\mathbf{x}'$ pose four degrees of freedom. Due to errors in image correspondence analysis, and the nature of numerical computations, back-projected rays never meet ideally at a point in 3D space. It is therefore required to define a to-be-minimised error metric for finding an optimal solution (X, Y, Z). There exists a number of methods to work out such a solution; see below.

Estimating the *uncertainty* of a solution is often as important as finding the solution itself. The uncertainty can be approximated by the propagation of the covariance matrix of $(\mathbf{x}, \mathbf{x}')$ through a first-order linearisation of the triangulation function τ. Let $\mathbf{\Sigma}_{\mathbf{x},\mathbf{x}'}$ be the 4×4 covariance matrix encoding the uncertainty of an image correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$. The error covariance of τ can be approximated as

$$\mathbf{\Sigma}_{\tau} \approx \mathbf{J}_{\tau} \mathbf{\Sigma}_{\mathbf{x},\mathbf{x}'} \mathbf{J}_{\tau}^{\top} \tag{3.2}$$

where

$$\mathbf{J}_{\tau} = \begin{pmatrix} \frac{\partial \tau}{\partial x} & \frac{\partial \tau}{\partial y} & \frac{\partial \tau}{\partial x'} & \frac{\partial \tau}{\partial y'} \end{pmatrix} \tag{3.3}$$

is the 3×4 *Jacobian matrix* of τ at $(\mathbf{x}, \mathbf{x}')$.

Method 1: Direct linear transform

The structure can be calculated in a linear manner if π and π' are perspective projections, as the perspective model linearly relates motion and structure in the projective space to each other.

Given a 3-by-4 projection matrix $\mathbf{P}$, a scene point (X, Y, Z) and its image (x, y), both in homogeneous coordinates, it follows that

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \mathbf{P} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 \end{pmatrix}^{\top} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{3.4}$$

where $\sim$ denotes the equality up to a scale. The scale ambiguity can be eliminated by dividing the first and the second rows of Eq. (3.4) by the third, which yields a homogeneous form

$$\begin{pmatrix} \mathbf{p}_1^\top - x\mathbf{p}_3^\top \\ \mathbf{p}_2^\top - y\mathbf{p}_3^\top \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{0}\,. \tag{3.5}$$

Note that the same equation can be obtained by applying a cross product with $(x, y, 1)$ on both sides of Eq. (3.4).

Taking into account observations from the second view, an over-determined homogeneous linear system is populated, having four equations constraining three unknowns:

$$\begin{pmatrix} \mathbf{p}_1^\top - x\mathbf{p}_3^\top \\ \mathbf{p}_2^\top - y\mathbf{p}_3^\top \\ \mathbf{p}_1'^\top - x'\mathbf{p}_3'^\top \\ \mathbf{p}_2'^\top - y'\mathbf{p}_3'^\top \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{A}\tilde{\mathbf{y}} = \mathbf{0}\,. \tag{3.6}$$

Let $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the *singular value decomposition* (SVD) of $\mathbf{A}$. The solution of Eq. (3.6) is found to be $\tilde{\mathbf{y}} = \mathbf{v}_4$, i.e. the 4-th right singular vector corresponding to the zero singular value (hence minimising $\|\mathbf{A}\tilde{\mathbf{y}}\|^2$). The Euclidean coordinates vector $\mathbf{y}$ is recovered by the conversion of $\tilde{\mathbf{y}}$ as follows:

$$\mathbf{y} = \left(\frac{\tilde{y}_1}{\tilde{y}_4}, \frac{\tilde{y}_2}{\tilde{y}_4}, \frac{\tilde{y}_3}{\tilde{y}_4} \right)^\top . \tag{3.7}$$

When the projection matrices are parametrized by upper triangular camera matrices $\mathbf{K}$ and $\mathbf{K}'$, we have that $\mathbf{P} = \mathbf{K}[\mathbf{I}\;\mathbf{0}]$ and $\mathbf{P}' = \mathbf{K}'[\mathbf{R}\;\mathbf{t}]$.

Let $\tilde{\mathbf{x}} = (x_1, x_2, 1)^\top$ be $\mathbf{x}$ in homogeneous coordinates, and $\pi_{\mathbf{K}}^{-1}(\mathbf{x}) = \mathbf{K}^{-1}\tilde{\mathbf{x}}$ the inverse[1] of the projection function $\pi_{\mathbf{K}}$. Using normalised image coordinates $\mathring{\mathbf{x}} = \pi_{\mathbf{K}}^{-1}(\mathbf{x})$

[1] The inverse of projection is also known as backward projection. As infinitely many points in 3D

and $\mathring{\mathbf{x}}' = \pi^{-1}_{\mathbf{K}'}(\mathbf{x}')$, the projection matrices become $\mathbf{P} = [\mathbf{I}\ \mathbf{0}]$ and $\mathbf{P}' = [\mathbf{R}\ \mathbf{t}]$.

Equation (3.6) can then be rewritten in non-homogeneous form as follows:

$$\mathbf{A}\mathbf{y} = \begin{pmatrix} 1 & 0 & -\mathring{x} \\ 0 & 1 & -\mathring{y} \\ r_{11} - r_{31}\mathring{x}' & r_{12} - r_{32}\mathring{x}' & r_{13} - r_{33}\mathring{x}' \\ r_{21} - r_{31}\mathring{y}' & r_{22} - r_{32}\mathring{y}' & r_{23} - r_{33}\mathring{y}' \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ t_3\mathring{x}' - t_1 \\ t_3\mathring{y}' - t_2 \end{pmatrix} = \mathbf{b}, \tag{3.8}$$

where r_{ij} is the entry at i-th row and j-th column of $\mathbf{R}$, and $\mathbf{t} = (t_1, t_2, t_3)^\top$. The triangulated 3D coordinates vector is then given by $\mathbf{y} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$. The use of normalised image coordinates has been found to improve the numerical stability (Hartley & Zisserman, 2004).

A generalisation of Eq. (3.6) into a multi-view case has a similar (low) complexity level as solving an over-determined homogeneous linear system. Estimating the structure in this way is known as *direct linear transform* (DLT). The DLT technique has been widely used not only for point triangulation but also for camera calibration, homography estimation, or pose recovery (Hartley & Zisserman, 2004). Equation (3.5) poses a linear duality between the structure and the motion if the number of constraints is sufficient. Thus, the motion can be immediately calculated in a least-squares form once the structure is determined, and vice versa. Such a duality allows the structure computation to be implicitly embedded into a pose recovery problem (Chien, Geng & Klette, 2015).

Applying the DLT method to find the structure $\mathbf{y}$ requires either inverting a 3×3 matrix, or finding the SVD of a 3×4 matrix, which are both computationally inexpensive. However, the solved structure minimises the algebraic distance $\delta_{\text{DLT}} = \|\mathbf{A}\mathbf{y}\|^2$ in homogeneous coordinates, hence it is geometrically meaningless (Hartley & Zisserman,

space can be projected onto the same image point, the inverse actually defines a line in the space, which is inherently modelled as a one-to-one function in homogeneous coordinates.

2004; Lindstrom, 2009). The remaining two methods in this section approach the problem of estimating structure in a Euclidean way.

Method 2: Mid-point triangulation

An alternative choice of an error metric is the shortest Euclidean length of a line connecting back-projected rays. In such a case, the error is defined with respect to free parameters $k, k' \in \mathbb{R}_+$ as follows:

$$\delta_{\mathrm{MID}}(k, k') = \|k\mathbf{R}\mathring{\mathbf{x}} + \mathbf{t} - k'\mathring{\mathbf{x}}'\|^2 \tag{3.9}$$

where $\mathring{\mathbf{x}} = \pi_{\mathbf{K}}^{-1}(\mathbf{x})$ and $\mathring{\mathbf{x}}' = \pi_{\mathbf{K}'}^{-1}(\mathbf{x}')$ are the directional vectors of the back-projected rays. The geometric interpretation of $\delta_{\mathrm{MID}}(k, k')$ is shown in Fig. 3.2. The minimum of Eq. (3.9) corresponds to the least-square solution of the following linear system:

$$\begin{pmatrix} -\mathbf{R}\mathring{\mathbf{x}} & \mathring{\mathbf{x}}' \end{pmatrix} \begin{pmatrix} k \\ k' \end{pmatrix} = \mathbf{A} \begin{pmatrix} k \\ k' \end{pmatrix} = \mathbf{t} \tag{3.10}$$

Let $m = \mathring{\mathbf{x}}'^{\top}\mathbf{R}\mathring{\mathbf{x}}'$, $n = \mathring{\mathbf{x}}^{\top}\mathring{\mathbf{x}}$ and $n' = \mathring{\mathbf{x}}'^{\top}\mathring{\mathbf{x}}'$ be the inner products of the directional vectors. The analytical solution

$$\begin{pmatrix} k \\ k' \end{pmatrix} = (\mathbf{A}^{\top}\mathbf{A})^{-1}\mathbf{A}^{\top}\mathbf{t} = \frac{1}{nn' - m^2} \begin{pmatrix} m\mathring{\mathbf{x}}'^{\top} - n'\mathring{\mathbf{x}}^{\top}\mathbf{R}^{\top} \\ n\mathring{\mathbf{x}}'^{\top} - m\mathring{\mathbf{x}}^{\top}\mathbf{R}^{\top} \end{pmatrix} \mathbf{t} \tag{3.11}$$

denotes two points on each of the back-projected rays at the shortest mutual distance in 3D space. The midpoint of those two points,

$$\mathbf{y} = \frac{1}{2}\left(k\mathbf{R}\mathring{\mathbf{x}} + \mathbf{t} + k'\mathring{\mathbf{x}}'\right) , \tag{3.12}$$

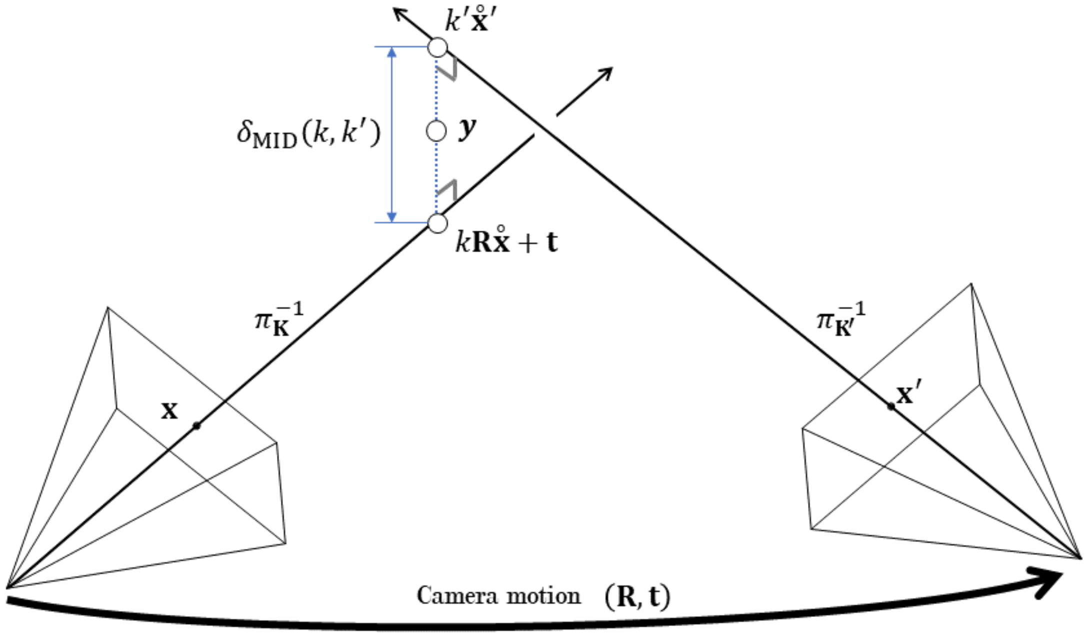

$k'\mathring{\mathbf{x}}'$
$\delta_{\mathrm{MID}}(k,k')$
$\mathbf{y}$
$k\mathbf{R}\mathring{\mathbf{x}}+\mathbf{t}$
$\pi_{\mathbf{K}}^{-1}$
$\pi_{\mathbf{K}'}^{-1}$
$\mathbf{x}$
$\mathbf{x}'$
Camera motion $(\mathbf{R},\mathbf{t})$

Figure 3.2: Geometry of mid-point triangulation. The dashed line segment perpendicular to the back-projected rays is the shortest line connecting them.

is therefore the optimal solution subject to the defined error metric. Substituting the values of k and k' from Eq. (3.11) into Eq. (3.12) shows the solution $\mathbf{y}$ in a plain form:

$$\mathbf{y} = \frac{1}{2}\left(\frac{m\left(\mathbf{M}-\mathbf{M}^{\top}\right)+n\mathbf{N}'-n'\mathbf{N}}{nn'-m^2}+\mathbf{I}\right)\mathbf{t} \tag{3.13}$$

where $\mathbf{M} = \mathbf{R}\mathring{\mathbf{x}}\mathring{\mathbf{x}}'^{\top}$, $\mathbf{N} = \mathring{\mathbf{x}}\mathring{\mathbf{x}}^{\top}$ and $\mathbf{N}' = \mathring{\mathbf{x}}'\mathring{\mathbf{x}}'^{\top}$ are the outer products of the directional vectors. This approach is known as *mid-point triangulation*.

The singularity of Eq. (3.13) happens when the back-projected rays are nearly parallel (i.e. $m \approx nn'$). A closed-form formulation of the partial derivatives is much more complicated than Eq. (3.13).

Compared to the DLT method, the mid-point triangulation is not only more computationally efficient but also more geometrically meaningful. The method can be generalised to multiple views (Beardsley, Zisserman & Murray, 1997).

Method 3: Optimal triangulation

An alternative way to evaluate the geometric fitness of a triangulated solution $\mathbf{y}$ is to measure the geodesic distances between its projection and observations $\mathbf{x}$ and $\mathbf{x}'$. For ensuring a projective invariant estimation, the problem is formulated as finding an optimal correspondence $\hat{\mathbf{x}} \leftrightarrow \hat{\mathbf{x}}'$ that is minimising the *reprojection error* (RPE) (Hartley & Sturm, 1995; Lindstrom, 2009; F. C. Wu, Zhang & Hu, 2011)

$$\delta_{\text{OPTIMAL}}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \|\mathbf{x}' - \hat{\mathbf{x}}'\|^2 \tag{3.14}$$

subject to the epipolar constraint $\hat{\mathbf{x}}'^{\top}\mathbf{F}\hat{\mathbf{x}} = 0$. The solution of Eq. (3.14) achieves a *maximum likelihood estimation* (MLE) if observations $\mathbf{x} \leftrightarrow \mathbf{x}'$ are corrupted by Gaussian noise.

As an optimal match $\hat{\mathbf{x}} \leftrightarrow \hat{\mathbf{x}}'$ is constrained to be on the epipolar line, a unique structure $\mathbf{y}$ is automatically found once Eq. (3.14) is minimised. In particular, the optimal structure is

$$\mathbf{y} = \frac{\mathbf{m}^{\top}\mathbf{E}\hat{\mathbf{x}}}{\mathbf{m}^{\top}\mathbf{m}}\hat{\mathbf{x}}' \tag{3.15}$$

where $\mathbf{m} = \mathbf{R}\hat{\mathbf{x}} \times \hat{\mathbf{x}}'$.

Solving this equation, however, poses a quadratically-constrained quadratic minimisation problem which, unfortunately, has no closed-form solution. Hartley and Sturm (Hartley & Sturm, 1995) re-parametrized the objective function using a variable that controls the pencil of epipolar lines; they suggested a numerical solver to find all roots of the resultant six-degree polynomial in terms of the introduced variable. Recently, various algorithms have been proposed to improve the numerical stability and computational cost in solving the optimal triangulation problem (Lindstrom, 2009; F. C. Wu et al., 2011).

The optimal solution can also be approached iteratively, using a numerical optimisation algorithm such as the one shown in Section 3.4.4. In this case, one may use the DLT technique or the mid-point method to obtain an initial solution. It is worth noting that, the minimisation of the reprojection error, as defined by Eq. (3.14), is considered the "gold standard" in the context of parameter estimation in computer vision. The objective is discussed repeatedly throughout this thesis (e.g. Sections 3.4.1, 4.3.2, and 5.4.2).

3.2.2 Disparity-depth conversion

In a special case where the image planes are co-planar and the cameras' principal axes point toward the same direction, the epipolar geometry is said to be *rectified* (Klette, 2014). As epipolar lines are parallel, one degree of freedom is eliminated from the second image point $\mathbf{x}'$. In particular, it follows that $\mathbf{x} = (x, y)^\top$ and $\mathbf{x}' = (x - d, y)^\top$, where $d \in \mathbb{R}_+$ is the displacement between corresponding pixels, known as *disparity*.

A stereo matcher provides a dense depth map by means of a (more or less) heuristic search in disparity space (Hirschmüller, 2008). For each pixel, a disparity is decided to achieve a minimum of a defined matching cost, specified by a data term and a smoothness constraint. Details can be found in Section 3.3.1.

A disparity-to-depth conversion transforms a pixel (x, y, d) into a point (X, Y, Z) in 3D space. This conversion can be linearly described in homogeneous coordinates as follows:

$$\tau(x, y, d) = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{b} & -\frac{x_c - x'_c}{b} \end{pmatrix} \begin{pmatrix} x \\ y \\ d \\ 1 \end{pmatrix}, \tag{3.16}$$

where b is the length of the baseline, (x_c, y_c) and (x'_c, y'_c) are the principle points of the rectified left and right camera, respectively, and f is the effective focal length after rectification (see Fig.3.3). More details of image rectification can be found in the text (Klette, 2014).

To study the error covariance of the conversion, we first abstract the perspective projection function and rewrite Eq. (3.16) in non-homogeneous form as follows:

$$\tau(x,y,d) = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \frac{bf}{d} \cdot \pi_{\mathbf{K}}^{-1}(x,y)\,, \tag{3.17}$$

where $\pi_{\mathbf{K}}^{-1}(x,y) = (\mathring{x}, \mathring{y}, 1)^\top$ is the back-projection of image coordinates (x,y) to a point $(\mathring{x}, \mathring{y}, 1)$ in the normalised image plane by the rectified camera matrix $\mathbf{K}$. Then we calculate the partial derivatives

$$\begin{aligned} \frac{\partial \tau}{\partial x} &= \frac{bf}{d} \cdot \frac{\partial \pi_{\mathbf{K}}^{-1}}{\partial x} &&= Z\left(\frac{\partial \mathring{x}}{\partial x}, \frac{\partial \mathring{y}}{\partial x}, 0\right)^\top \\ \frac{\partial \tau}{\partial y} &= \frac{bf}{d} \cdot \frac{\partial \pi_{\mathbf{K}}^{-1}}{\partial y} &&= Z\left(\frac{\partial \mathring{x}}{\partial x}, \frac{\partial \mathring{y}}{\partial x}, 0\right)^\top \\ \frac{\partial \tau}{\partial d} &= -\frac{bf}{d^2} \cdot \pi_{\mathbf{K}}^{-1} &&= Z\left(-\frac{\mathring{x}}{d}, -\frac{\mathring{y}}{d}, -\frac{1}{d}\right)^\top \end{aligned} \tag{3.18}$$

In the rectified case it holds that $\frac{\partial \mathring{x}}{\partial y}(x,y) = \frac{\partial \mathring{y}}{\partial x}(x,y) = 0$ and $\frac{\partial \mathring{x}}{\partial x}(x,y) = \frac{\partial \mathring{y}}{\partial y}(x,y) = \frac{1}{f}$. In this case, the Jacobian matrix of τ is reduced into an upper triangular form

$$\mathbf{J}_\tau(x,y,d) = Z \cdot \begin{pmatrix} \frac{1}{f} & 0 & -\frac{\mathring{x}}{d} \\ 0 & \frac{1}{f} & -\frac{\mathring{y}}{d} \\ 0 & 0 & -\frac{1}{d} \end{pmatrix}, \tag{3.19}$$

which propagates error covariance from the image-disparity space to the 3D Euclidean

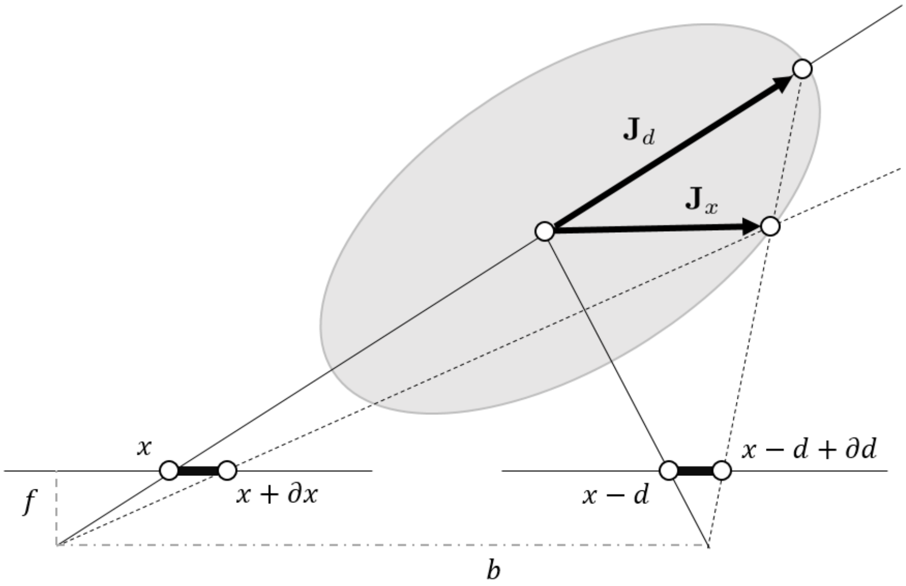


Figure 3.3: The shaded area shows the 1-σ ellipsoid corresponding to Σ_τ on the X-Z plane. The vectors $\mathbf{J}_d$ and $\mathbf{J}_x$ are the third and first column of $\mathbf{J}_\tau$, respectively.

space by

$$\Sigma_\tau(x, y, d) = \mathbf{J}_\tau \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_d^2 \end{pmatrix} \mathbf{J}_\tau^\top \tag{3.20}$$

where σ_x, σ_y, and σ_d are the standard deviations with respect to the independent variables x, y, and d, respectively.

A geometric representation of Eq. (3.20) as a 1-σ error ellipsoid, with σ_x and σ_d set to 1, is visualised in Fig. 3.3. Figure 3.4 shows a point cloud in 3D space reconstructed by using disparity-to-depth An in-depth error analysis of disparity-generated 3D data can be found in (Kim, Ansar, Steele & Steinke, 2005).

3.3 Establishing image correspondence

Finding image correspondences plays a key role in a variety of vision tasks, and it is no exception in the field of VO and SLAM. Correspondences between observations of

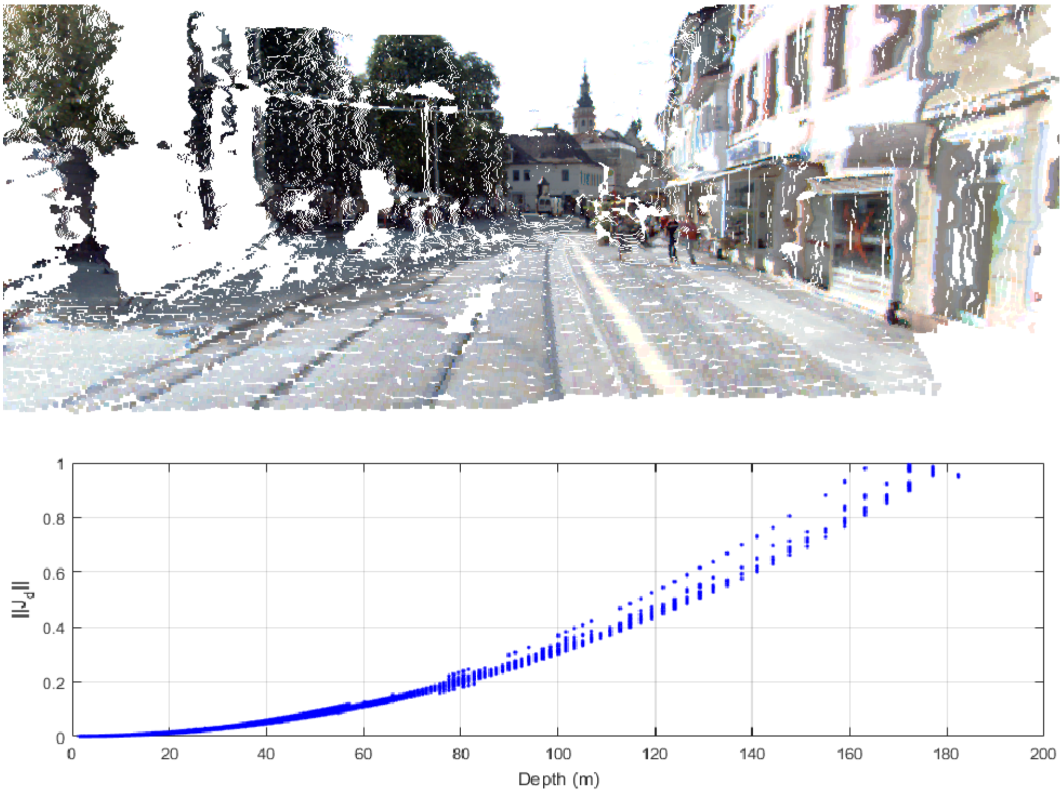


Figure 3.4: The upper plot shows a scene reconstructed from a disparity map on the X-Z plane with having the Y-coordinate colour coded. The bottom plot visualises the halved length of the principal axis of each reconstructed point's error ellipsoid following Eq. (3.19), showing that the uncertainty of triangulation grows nonlinearly with respect to depth. The magnitude of $\mathbf{J}_d$ at depth Z = 120 metres is about 0.5 metre. The figures are rendered from a frame from sequence `2011_09_26_drive_0091_sync` of the KITTI Vision Benchmark Suite (Geiger et al., 2013).

Table 3.2: Comparison of two basic options for establishing image correspondences.

	Photometric matching	**Descriptor matching**
Domain	Image intensity	Vectors in feature space
Transform invariant	No	Feature-space dependent
Epipolar constraint	Preserved	Ignored *
Point density	Sparse or dense	Sparse **
Image patch	Sharp, blurred or textureless	Sharp texture required
Point displacement	Limited to a search range	Unlimited in feature space

* Post-filtering techniques such as RANSAC are required.
** Dense correspondences are possible but at a very high computational cost and with poor accuracy.

a stationary scene point in multiple views are useful to derive the 3D coordinates of this point (as discussed in Section 3.2) and the egomotion of the camera (covered in Section 3.4). Therefore, image correspondences serve as a bridge between structure and motion.

This section describes two classes of techniques for establishing image correspondences in subsequently recorded images.

Methods in the first class of *photometric matching* perform direct matching on image data; this is useful for generating dense correspondences subject to a global smoothness regularisation. Methods in the second class of *descriptor matching* find sparse image correspondences by matching characteristic vector representations of sparse key points, known as descriptors in a feature space. Both classes are compared in Table 3.2.

3.3.1 Photometric matching

A straightforward way to establish correspondences between two subsequently recorded images is to perform direct matching on pixel intensities. The general intensity-based matching problem, regarding two temporally related images, is known as *optical flow estimation*.

This field has been extensively studied in computer vision. Many approaches such as the Horn-Schunck (Horn & Schunck, 1981) algorithm, the Lucas-Kanade algorithm (Lucas & Kanade, 1981; Baker & Matthews, 2004), and the Farnebäck algorithm (Farnebäck, 2003), have been developed and successfully applied to solve a wide variety of problems. The Lucas-Kanade algorithm was later further developed into a point tracker (Tomasi & Kanade, 1991; Shi & Tomasi, 1993), known as the *KLT method*, which has been adopted by many modern VO implementations (Badino, Yamamoto & Kanade, 2013; Ci & Huang, 2016).

The KLT method performs correspondence search along an image gradient-guided path which might violate epipolar constraints, even for a static 3D point. It is therefore more appropriate to constrain the search space on the epipolar line when the camera motion is known. This way not only the epipolar condition is preserved but also the dimensionality of the search space is greatly reduced from 2D to 1D.

To perform a correspondence search on an epipolar line, defining a way to parameterise a correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ is required. *Inverse depth parameterization* (Civera, Davison & Montiel, 2008) is a commonly adopted approach in the context of SLAM. Using the inverse depth of $\mathbf{x}$, its 3D coordinates are defined in the reference frame as follows:

$$g(\mathbf{x}, d) = \frac{1}{d}\pi^{-1}(\mathbf{x}) \tag{3.21}$$

where d is the inverse depth parameter. The correspondence hypothesis in the second view is therefore

$$\mathbf{x}' = \pi'\left(\mathbf{R}g(\mathbf{x}, d) + \mathbf{t}\right) \tag{3.22}$$

where $[\mathbf{R}, \mathbf{t}]$ denotes the pose of the reference frame with respect to the base.

To find the corresponding point $\mathbf{x}'$ that best matches $\mathbf{x}$, given a reference image I', a photometric error function is defined:

$$C(\mathbf{x}, d) = \delta\left[I\left(\mathbf{x}\right) - I'\left(\mathbf{x}'\right)\right] \tag{3.23}$$

where $\delta : \mathbb{R} \to \mathbb{R}_+$ is a selected metric and $\mathbf{x}'$ is the correspondence hypothesis generated from d "on-the-fly". The best match is found to maximise the photoconsistency, specified as follows:

$$\hat{d} = \underset{d \in \mathbb{D}}{\operatorname{argmin}} \left\{C(\mathbf{x}, d)\right\} \tag{3.24}$$

When an estimate $(\hat{d}_0, \sigma_d)$ is known beforehand, the search space can be further bounded to a portion on the epipolar line, based on a confidence interval, say $\mathbb{D} = [\hat{d}_0 - \sigma_d, \hat{d}_0 + \sigma_d]$, if it is known, or an empirically set value.

In practice, instead of using single pixel intensities, a block of neighbouring pixels is taken into account to improve the matching accuracy and robustness. Patch-based photoconsistency is defined by

$$C(\mathbf{x}, d) = \delta\left[\mathcal{I}\left(\mathbf{x}\right) - \mathcal{I}'\left(\mathbf{x}'\right)\right] \tag{3.25}$$

where $\mathcal{I}(\mathbf{x})$ samples a block of $W \times W$ image intensities around centre pixel $\mathbf{x}$ and $\delta : \mathbb{R}^{W \times W} \times \mathbb{R}^{W \times W} \to \mathbb{R}$ is any suitable metric that measures photoconsistency between two blocks.

Frequently used metrics are the *sum-of-squared-differences* (SSD), the *sum-of-absolute-differences* (SAD), cross-correlation, or *mutual information* (MI) (Scharstein & Szeliski, 2002). The window can be optionally warped using a perspective transform at an extra computational cost (Forster, Pizzoli & Scaramuzza, 2014).

When it comes to establishing dense correspondences, it is often desired to regularise the per-pixel local search in Eq. (3.24) to maintain global smoothness. Regularised search leads to the problem of finding a dense map D that minimises the functional

$$\Phi(D) = \sum_{\mathbf{x}\in\Omega} \left[C\left(\mathbf{x}, D(\mathbf{x})\right) + \lambda \sum_{\mathbf{u}\in A_{\mathbf{x}}} E(\mathbf{x}, \mathbf{u}; D) \right] \tag{3.26}$$

where Ω is the domain of image I, $A_{\mathbf{x}}$ is a set of neighbouring pixels around $\mathbf{x}$, E is a pair-wise smoothness penalty function, and $\lambda > 0$ is the damping variable controlling the importance.

The minimisation of Φ has been well studied in the special case where the first and second views form a rectified left-right stereo-image pair. Some stereo matchers, such as the *semi-global matcher* (SGM) (Hirschmüller, 2008), for example, have been developed to compute an optimal map D. Note that the inverse depth is proportional to the disparity value (in the rectified case) by factor bf, where b is the length of the stereo baseline, and f is the effective focal length.

Block matching can further be generalised to multiple views (Newcombe, Lovegrove & Davison, 2011). Given a set of N reference frames $\{I_j\}$, each of which is equipped with a transform $\mathbf{T}_j$ from the base frame, the data cost in Eq. (3.26) is extended into a multi-view variant:

$$C(\mathbf{x}, d) = \sum_{1\le j\le N} \delta\left[\mathcal{I}\left(\mathbf{x}\right), \mathcal{I}_j\left(\pi\left(\mathbf{T}_j\, \tilde{g}(\mathbf{x}, d)\right)\right)\right]. \tag{3.27}$$

As the computation is very expensive, it is often done in an off-line phase.

3.3.2 Descriptor matching

Photometric matching methods require a known camera pose to actualise the perspective warping. When this is not the case, descriptor-based matching offers an alternative solution. A descriptor $\nu(\mathbf{x})$ is the vector representation of an image point $\mathbf{x}$ derived from image I using a feature transform (Klette, 2014). Instead of searching in the image domain, point correspondences are established in the feature space.

Let $\mathcal{F}$ and $\mathcal{F}'$ be sets of image features identified in the first and second views. A feature point $\chi \in \mathcal{F}$ is mapped onto $\chi' \in \mathcal{F}'$ where

$$\chi' = \operatorname*{argmin}_{\chi^\star \in \mathcal{F}'} \delta \left[\nu(\chi), \nu(\chi^\star) \right] \tag{3.28}$$

using a metric δ that measures the similarity among two feature vectors. In the case of binary descriptors, the *Hamming distance* is used; otherwise one might choose SSD or SAD as a metric.

Pairwise nearest neighbour matching can be done by using either a brute-force search over $\mathcal{F}'$, or by performing a heuristic strategy (Muja & Lowe, 2009).

When features are extracted from an image with repetitive patterns or similar textures, then there may exist an ambiguity in the resulting matching. Ambiguous matches are identified by a difference ratio check. Let $\chi'' \in \mathcal{F}'$ be the second best match of χ. The mapping $\chi \rightarrow \chi'$ is said to be ambiguous if

$$\frac{\delta(\chi, \chi')}{\delta(\chi, \chi'')} < R \tag{3.29}$$

where $0 \leq R < 1$ is a pre-defined threshold for this distance ratio. In the literature it is suggested to set $R = 0.8$ (Lowe, 2004).

It is also important to remove inconsistent matches by performing a process of backward matching, from $\mathcal{F}'$ to $\mathcal{F}$. In particular, a match $\chi \rightarrow \chi'$ is considered to be inconsistent if it is found that

$$\operatorname*{argmin}_{\chi^\star \in \mathcal{F}} \delta \left[\nu(\chi'), \nu(\chi^\star) \right] \neq \chi \,. \tag{3.30}$$

After enforcing Eqs. (3.29) and (3.30), a set of unambiguously symmetric matches

$\{\chi \leftrightarrow \chi'\}$ is established. Correspondence search in feature space is able to identify corresponding points with large displacements in two images which is often a challenging case in photometric methods.

Feature-based correspondences, however, may violate the epipolar constraint, as image topology is not preserved in feature space. It is necessary to conduct filtering techniques before the established correspondences can be used for egomotion estimation (Fraundorfer & Scaramuzza, 2012). A variety of geometric filtering approaches are detailed in Section 4.2.

3.4 Camera motion recovery

Motion recovery is one of the key tasks of a mapper or odometer. The motion of a camera can be estimated by using correspondences between 2D, 3D, or photometric spaces, considering these spaces for two views. In this section we provide a brief initial review of widely adopted two-view pose estimation techniques. A more comprehensive study is later given in Section 4.3.

3.4.1 Perspective-n-point (PnP) problem

Pose recovery of a camera, given a set of 3D points in the first view and their projections in the second, is known as the *perspective-n-point* (PnP) problem. The problem can be linearly solved following a DLT technique, similar to the one previously described in Section 3.2.1.

Let $\{\mathbf{y}_i\}$ be a set of 3D points observed in the first view, and $\{\mathbf{x}'_i\}$ be the set of corresponding image points in the second view. For each correspondence we have that

$$\tilde{\mathbf{x}}'_i = \begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix} \sim \mathbf{P}\tilde{\mathbf{y}}_i \tag{3.31}$$

where $\mathbf{P} = \mathbf{K}[\mathbf{R} \;\; \mathbf{t}]$ is the 3×4 projection matrix containing known camera matrix $\mathbf{K}$ and unknowns $\mathbf{R}$ and $\mathbf{t}$.

If the matrix $\mathbf{P}$ is treated as a "black box", the correspondence forms a homogeneous linear system of twelve unknowns

$$\begin{pmatrix} \tilde{\mathbf{y}}_i^\top & \mathbf{0} & -x'_i\tilde{\mathbf{y}}_i^\top \\ \mathbf{0} & \tilde{\mathbf{y}}_i^\top & -y'_i\tilde{\mathbf{y}}_i^\top \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{0} \tag{3.32}$$

where $\mathbf{p}_j^\top$ is the j-th row of $\mathbf{P}$.

By stacking the constraints of more than six correspondences $\mathbf{x}_i \leftrightarrow \mathbf{y}_i$, the matrix $\mathbf{P}$ can be uniquely solved based on

$$\begin{pmatrix} \tilde{\mathbf{y}}_1^\top & \mathbf{0} & -x'_1\tilde{\mathbf{y}}_1^\top \\ \mathbf{0} & \tilde{\mathbf{y}}_1^\top & -y'_1\tilde{\mathbf{y}}_1^\top \\ \tilde{\mathbf{y}}_2^\top & \mathbf{0} & -x'_2\tilde{\mathbf{y}}_2^\top \\ \mathbf{0} & \tilde{\mathbf{y}}_2^\top & -y'_2\tilde{\mathbf{y}}_2^\top \\ \vdots & \vdots & \vdots \\ \tilde{\mathbf{y}}_6^\top & \mathbf{0} & -x'_6\tilde{\mathbf{y}}_6^\top \\ \mathbf{0} & \tilde{\mathbf{y}}_6^\top & -y'_6\tilde{\mathbf{y}}_6^\top \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{0} \tag{3.33}$$

using the aforementioned SVD approach followed by a normalisation having $p_{34} = 1$.

The pose can then be recovered by $[\mathbf{R}\ \ \mathbf{t}] = \mathbf{K}^{-1}\mathbf{P}$.

The over-parametrized rotation matrix $\mathbf{R}$, solved in a linear way, needs to be projected into $\mathbb{SO}(3)$ to be a valid orthonormal matrix representation for rotation.

Furthermore, the linear technique is known to be unstable when the correspondences $\{\mathbf{x}'_i \leftrightarrow \mathbf{y}_i\}$ are noisy. Various sophisticated solvers have been proposed to address these issues. The recently proposed *efficient PnP solver* (EPnP) (Lepetit, Moreno-Noguer & Fua, 2009) is considered to be one of the state-of-the-art approaches to deal with a large number of noisy correspondences in linear time complexity (Mur-Artal et al., 2015).

The success of Lepetit et al.'s solver is based on a reformulation of Eq. (3.31) as a four-point rigid alignment problem, which has a trivial closed-form solution. It first selects four reference points $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$ that form a simplex, containing all the 3D points $\{\mathbf{y}_i\}$, each of which are then transformed into barycentric coordinates $\mathbf{a}_i = (a_{i1}, a_{i2}, a_{i3}, a_{i4})^\top$ in the subspace spanned by the simplex, following the mapping

$$\mathbf{y}_i = \sum_{j=1}^{4} a_{ij}\mathbf{c}_i = \mathbf{\Gamma}\mathbf{a}_i \tag{3.34}$$

where $\mathbf{\Gamma}$ is the 3×4 matrix with $\mathbf{c}_j$ in its j-th column.

The projection of $\mathbf{y}_i$ in the first view now becomes

$$\tilde{\mathbf{x}}_i \sim \mathbf{K}\mathbf{y}_i = \mathbf{K}\mathbf{\Gamma}\mathbf{a}_i\,. \tag{3.35}$$

Let $\mathbf{\Gamma}' = [\mathbf{c}'_1\ \ \mathbf{c}'_2\ \ \mathbf{c}'_3\ \ \mathbf{c}'_4]$ be the matrix of reference points in the second view, with the transform $[\mathbf{R}\ \ \mathbf{t}]$ applied column-wise:

$$\mathbf{c}'_j = \mathbf{R}\mathbf{c}_j + \mathbf{t}, \quad j = 1, 2, 3, 4\,. \tag{3.36}$$

Since the barycentric coordinates are invariant under rigid transforms, the projection in the second view can be rewritten as

$$\tilde{\mathbf{x}}'_i \sim \mathbf{K}\mathbf{\Gamma}'\mathbf{a}_i = \mathbf{P}'\mathbf{a}_i \tag{3.37}$$

where an “agent” projection matrix $\mathbf{P}'$ has been introduced in this equation.

The matrix can be solved by using the linear technique previously applied to Eq. (3.33). Once $\mathbf{P}'$ is found, $\mathbf{\Gamma}'$ can be recovered projectively (i.e. $\mathbf{\Gamma}' \sim \mathbf{K}^{-1}\mathbf{P}'$).

It is straightforward to find the absolute scale from $\mathbf{\Gamma}$, as rigid transforms are norm-preserving. The last step is to extract $\mathbf{R}$ and $\mathbf{t}$ from the correspondences $\{\mathbf{c}_j \leftrightarrow \mathbf{c}'_j\}$ according to Eq. (3.36). This can be efficiently done using quaternions (Horn, 1987). Details regarding rigid alignment are discussed in Section 4.3.4.

3.4.2 Essential matrix decomposition

Under some circumstances, such as monocular visual odometry, 3D coordinates $\mathbf{y}$ might not be available as a prior. In such a case, the motion of the camera can still be recovered from epipolar conditions but where the scale of $\mathbf{t}$ remains undetermined due to lack of metric reference (Longuet-Higgins, 1981).

Without loss of generality, we assume that $\|\mathbf{t}\| = 1$ in the following. Let $\mathbf{x} \leftrightarrow \mathbf{x}'$ be a 2D-to-2D correspondence. It follows that

$$\tilde{\mathbf{x}}'^{\top}\mathbf{K}^{-\top}[\mathbf{t}]_{\times}\mathbf{R}\mathbf{K}^{-1}\tilde{\mathbf{x}} = 0 \tag{3.38}$$

where

$$[\mathbf{t}]_{\times} = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix} \tag{3.39}$$

denotes the skew-symmetric form of $\mathbf{t} = (t_1, t_2, t_3)^{\top}$. Equation (3.38) is the well-known epipolar constraint , and the matrix $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$ is called the *essential matrix* (Hartley & Zisserman, 2004).

The essential matrix can be recovered without knowing any 3D structure. Among a variety of essential matrix recovery techniques, the *eight-point algorithm* is a popular choice (Chojnacki & Brooks, 2003). This algorithm first estimates the *fundamental matrix* $\mathbf{F} = \mathbf{K}^{-\top}\mathbf{E}\mathbf{K}^{-1}$ using at least eight point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$. For each correspondence, a homogeneous constraint is introduced by Eq. (3.38) as follows:

$$x_i x'_i f_{11} + y_i x'_i f_{12} + x'_i f_{13} + x_i y'_i f_{21} + y_i y'_i f_{22} + y'_i f_{23} + x_i f_{31} + y_i f_{32} + f_{33} = 0 \quad (3.40)$$

where f_{jk} denotes the entry of the j-th row and k-th column of the fundamental matrix.

By means of the SVD technique, all the nine elements of the fundamental matrix can be determined (up to a scale) from at least eight constraints. This is known as the *eight-point algorithm* in computer vision (Longuet-Higgins, 1981; Hartley, 1997).

According to $\mathbf{E} = \mathbf{K}^\top \mathbf{F}\mathbf{K}$, the essential matrix is recovered from the solved fundamental matrix. To extract pose $[\mathbf{R} \;\; \mathbf{t}]$ from an essential matrix $\mathbf{E}$, one may decompose it using the SVD technique:

$$\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^\top \quad (3.41)$$

where $\mathbf{U}$ and $\mathbf{V}$ are 3×3 orthonormal matrices, and $\mathbf{D} = \text{diag}(1, 1, 0)$ is a diagonal matrix having 1 at the first and second diagonal element, and 0 at the third (due to the rank deficiency of $\mathbf{E}$).

By introducing two matrices

$$\mathbf{Z} = \begin{bmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} 0 & \mp 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}, \quad (3.42)$$

and based on $\mathbf{D} = \mathbf{ZW}$ and $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, we can rewrite Eq. (3.41) as follows:

$$\mathbf{E} = \mathbf{UZU}^\top\mathbf{UWV}^\top . \tag{3.43}$$

It is verified that $\mathbf{S} = \mathbf{UZU}^\top$ is a skew-symmetric matrix, and $\mathbf{R}' = \mathbf{UWV}^\top$ is an orthonormal matrix. Following the definition $\mathbf{E} = [\mathbf{t}]_\times\mathbf{R} = \mathbf{SR}'$, the rotation matrix $\mathbf{R} = \mathbf{R}'$ and the unit translation vector $\mathbf{t} = (S_{32}, S_{13}, S_{21})^\top$ are instantly found.

Due to the sign ambiguity of $\mathbf{Z}$ and $\mathbf{W}$, there are four possible solutions. As described in the next section, the best candidate is decided by applying a triangulation method on $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, and checking the number of resulting points that fall *in front of* the cameras to select the best candidate. In the non-singular case, only one candidate gives a valid geometric setup. Figure 3.5 depicts an example of all the four possible solutions.

As an essential matrix has five degrees of freedom (three from rotation, two from translation with scale ambiguity), obviously, solving it using eight-point correspondences is an over-parametrized approach. It has been shown that the essential matrix can be recovered efficiently by using minimally five image correspondences to find the roots of a degree-10 univariate polynomial, which is known as the *five-point algorithm* (Nistér, 2004). The algorithm is further improved by Hartley et al. and now considered as a state-of-the-art solver (Li & Hartley, 2006).

3.4.3 Correspondence-free methods

When the scene structure is known, it is possible to determine the pose without any pre-computed correspondences which, in fact, will be established on-the-fly while the pose is being solved. A well-known method in this category is the *iterative-closest point* (ICP) algorithm (Besl & McKay, 1992). The technique is useful to align two sets of point clouds, which are, in the context of two-view pose recovery, the reconstructed

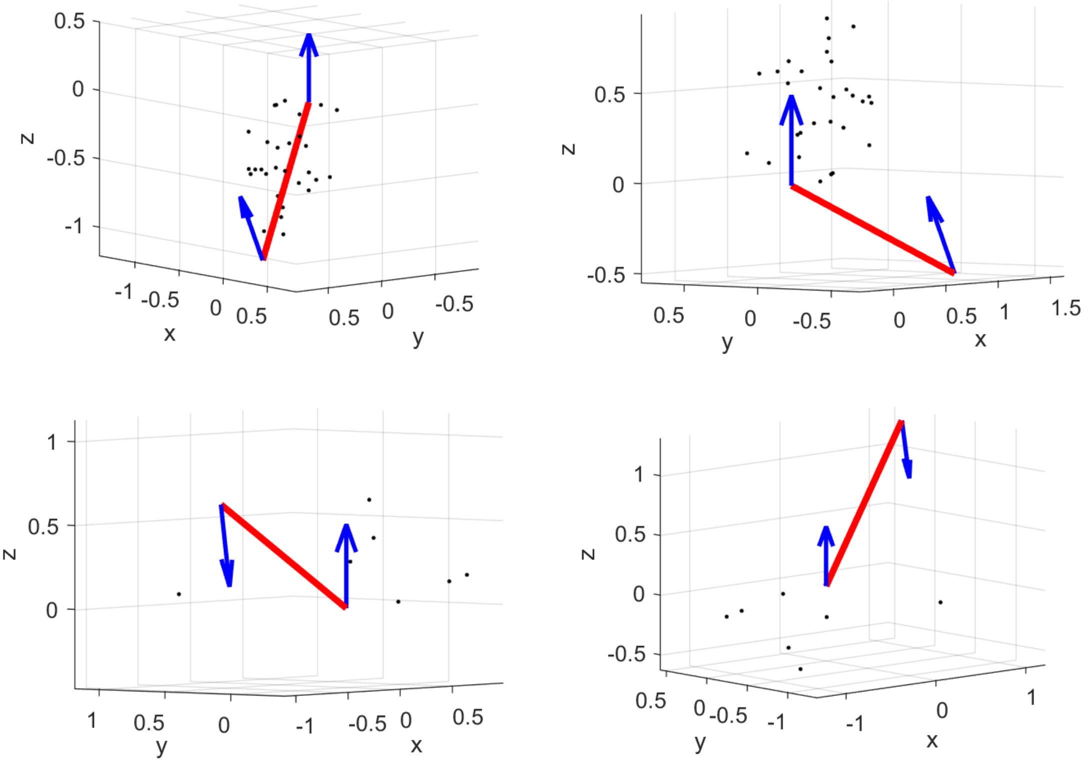


Figure 3.5: An example of four possible egomotion estimates from essential matrix decomposition. Each arrow starts from the centre of projection and points to the orientation of view. Only the top-right one shows a valid geometric configuration where all the triangulated 3D points lie in front of both cameras.

scene structure in the first and in the second view. However, the reconstruction from a disparity image is often too noisy to yield a good 3D alignment using ICP (Scaramuzza & Fraundorfer, 2011; Steinbruecker, Sturm & Cremers, 2011; Ci & Huang, 2016).

To overcome the inaccuracy of the structure estimate, mainstream correspondence-free methods use not only depth data but also the source image. These methods are designed to search for a rigid transform $(\mathbf{R}, \mathbf{t})$ such that, once applied for warping the image from the first view perspectively to the second using the depth as prior, the photometric difference is minimised. In particular, the estimation is based on the minimisation of the photoconsistency error previously defined by Eq. (3.23) in Section 3.3.1, where the scene structure is now considered as a known factor, and the transform $\mathbf{T}$ as being the unknown.

An example of such a photometric-driven adjustment is illustrated by Fig. 3.6. As the estimation is completely driven by image data, there exists no closed-form solution. Instead, an iterative nonlinear optimisation needs to be carried out. The minimisation process is detailed in Section 3.4.4.

This approach has been widely adopted by many real-time direct VO implementations that avoid feature extraction, matching, and filtering due to prohibited computational cost (Newcombe et al., 2011; Engel et al., 2013; Forster et al., 2014). This approach is also considered being the standard VO approach for RGB-D cameras (Steinbruecker et al., 2011; Kerl, Sturm & Cremers, 2013).

A direct search of optimal pose in the image space, however, can be unreliable when significantly many pixels are occluded, moving, largely displaced, or displaying non-Lambertian reflectance. Under such circumstances the alignment may diverge. We visit the photometric alignment issue again in Section 4.3.3.

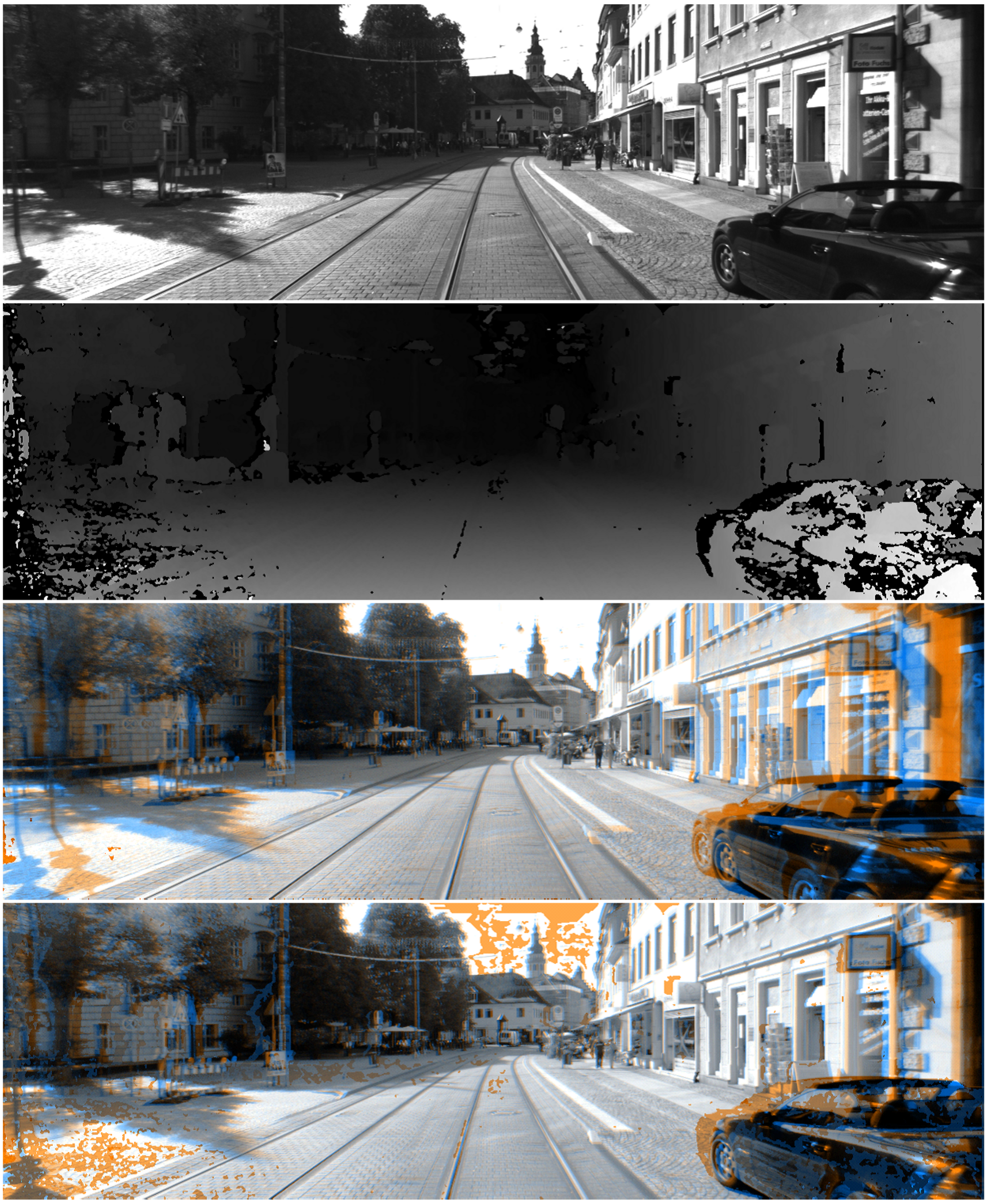


Figure 3.6: An example of pose recovery using direct photometric alignment. The top two rows show the target image and the source disparity map computed from the source stereo image (not displayed). Third row and fourth row visualise overlapped perspective warping of the source image to the target frame, using camera motion, respectively, before and after the optimisation. The mis-alignment of the closest store sign and windows on the right is significantly removed after the adjustment. The figures are rendered from the same frame shown in Fig.3.4.

3.4.4 Iterative energy minimisation

A nonlinear optimisation process is usually carried out to refine a linearly solved pose (e.g. EPnP), or to solve the pose if a closed-form solver is not available (e.g. direct photometric alignment). In this section we formulate an abstract framework to treat the pose estimation problem as an energy minimisation process.

So far this chapter represented a camera pose as a Euclidean transform $\mathbf{T} \in \mathbb{SE}(3)$ that consists of a rotation matrix $\mathbf{R} \in \mathbb{SO}(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. It is known that a Euclidean transform in 3D space has 6 degrees of freedom, while there are, however, twelve entries in $[\mathbf{R} \;\; \mathbf{t}]$.

To study pose recovery as an optimisation problem, a minimal representation must be adopted. One of those representations is to use twist coordinates $\xi \in \mathbb{R}^6$, which is a Lie-algebra entity minimally representing a Euclidean transform. Converting a twist to and from its corresponding Euclidean transform follows

$$\mathbf{T} = \exp_{\mathfrak{se}(3)}(\xi) \;\text{ and }\; \xi = \log_{\mathbb{SE}(3)}(\mathbf{T}) \tag{3.44}$$

where $\mathfrak{se}(3)$ is the Lie algebra corresponding to the Lie group $\mathbb{SE}(3)$. Two twists ξ and ξ' can be composited by the multiplication of their corresponding Euclidean transform matrices

$$\xi \circ \xi' = \log_{\mathbb{SE}(3)}\left(\exp_{\mathfrak{se}(3)}(\xi') \cdot \exp_{\mathfrak{se}(3)}(\xi)\right) \tag{3.45}$$

where $\circ$ is the pose concatenation operator.

To formulate pose recovery as an energy minimisation problem, a residual function $\varphi(\mathbf{x}, \mathbf{y}; \xi) \in \mathbb{R}$, parametrized over ξ, is defined for each established correspondence $\mathbf{x} \leftrightarrow \mathbf{y}$ to solve for pose. Note that $\mathbf{x}$ and $\mathbf{y}$ can be any entities of interest. Individual residuals are further summarised as a scalar to be minimised. This is commonly done in the sum-of-squares form to achieve a *maximum-likelihood estimation* (MLE) when the error distribution of residuals is believed to follow a Gaussian.

Let $\Phi(\xi) = (\varphi_1, \varphi_2, ..., \varphi_N)^\intercal$ be an N-vector function instantiated from N correspondences. The optimal pose estimate is found to be

$$\hat{\xi} = \operatorname*{argmin}_{\xi \in \mathbb{R}^6} \|\Phi(\xi)\|^2_{\mathbf{\Sigma}} \tag{3.46}$$

where $\|\cdot\|^2_{\mathbf{\Sigma}}$ is the squared *Mahalanobis distance* defined by $\mathbf{\Sigma}$, an $N \times N$ positive-definite matrix denoting the error covariance over all the correspondences.

When the correspondences are believed to be established independently (as in most of the cases), $\mathbf{\Sigma}$ is simplified as a diagonal matrix. Equation (3.46) can then be rewritten as

$$\hat{\xi} = \operatorname*{argmin}_{\xi \in \mathbb{R}^6} \sum_i w_i \left\|\varphi_i\left(\mathbf{x}_i, \mathbf{y}_i; \xi\right)\right\|^2 \tag{3.47}$$

where w_i is the inverse of the i-th diagonal entry in $\mathbf{\Sigma}$. An optimal estimate $\hat{\xi}$, that minimises the weighted sum-of-squares, can be approached iteratively by

$$\hat{\xi}_{k+1} = \Delta\xi_k \circ \hat{\xi}_k \tag{3.48}$$

with the update computed using the Levenberg-Marquardt algorithm (Levenberg, 1944):

$$\Delta\xi_k = \left(\mathbf{H} + \lambda \operatorname{diag}(\mathbf{H})\right)^{-1} \mathbf{J}^\intercal \Phi(\hat{\xi}_k) \tag{3.49}$$

where $\lambda \in \mathbb{R}$ is the damping variable, and $\mathbf{H} = \mathbf{J}^\intercal \mathbf{W} \mathbf{J}$ is the Hessian matrix approximated by the weight matrix $\mathbf{W} = \operatorname{diag}(w_0, w_1, ..., w_{i-1})$ and the Jacobian

$$\mathbf{J}_{ij} = \frac{\partial \varphi_i}{\partial \xi_j}(\hat{\xi}_k) \tag{3.50}$$

of Φ at $\hat{\xi}_k$.

The variable λ is adaptively adjusted to control the optimisation toward a Gauss-Newton-like process (when ξ is far from a local minimum), or a gradient-descent-like process (when ξ is closer to a local minimum.) The models, considered in this work, are minimised in this manner with the Jacobian matrix numerically computed by first-order finite differentiations. The next chapter discusses a variety of energy models; see Section 4.3.

3.5 Closure

This chapter provided a walk-through in three key areas of multi-view geometry, namely scene structure, image correspondences, and camera motion. The mathematical models, connecting those areas, serve as building blocks to develop a generic multi-objective visual odometry algorithm in the next chapter.

Chapter 4

Visual Odometry

Visual odometry tracks projections of a set of landmarks in the scene, frame-by-frame, and continuously derives the camera's egomotion. The historical path of developments in VO has lead to different models and implementation details. In this chapter we propose a novel formulation unifying these models to achieve accurate egomotion estimation while maintaining a good flexibility that also supports extensions to a variety of sensor configurations.

4.1 Overview

Let $I_j^k : \Omega_k \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ denote the image captured in frame j by camera k where Ω_k is the camera's image domain, and $\mathbf{T}_j$ the Euclidean transform from a reference frame to frame j.

A K-camera VO problem is to derive a sequence of transforms $(\mathbf{T}_1, \mathbf{T}_2, \ldots \mathbf{T}_N)$, given a sequence of N images $(I_1^k, I_2^k, \ldots I_N^k)$, where $1 \leq k \leq K$.

Typically the problem is approached by subsequently solving a series of two-frame pose estimation problems, each of which yields an instantaneous transform $\Delta\mathbf{T}_j$, from

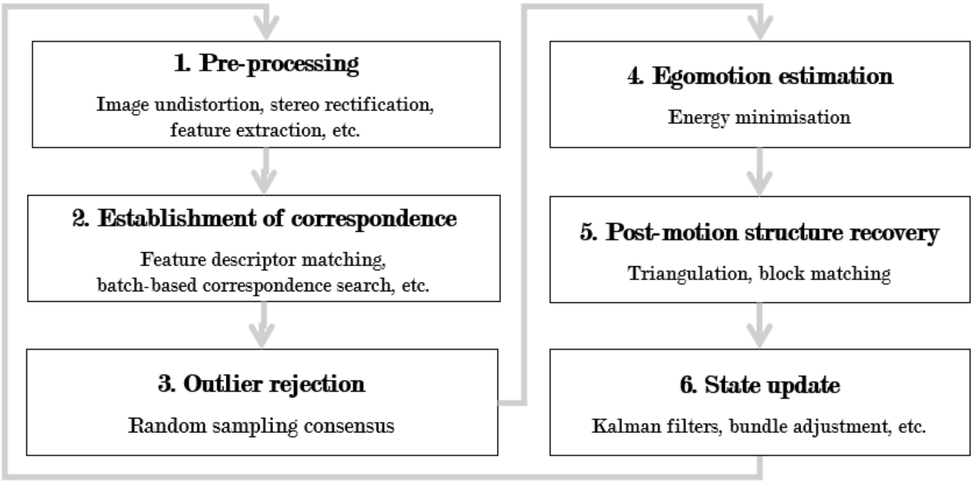

1. Pre-processing
Image undistortion, stereo rectification, feature extraction, etc.
2. Establishment of correspondence
Feature descriptor matching, batch-based correspondence search, etc.
3. Outlier rejection
Random sampling consensus
4. Egomotion estimation
Energy minimisation
5. Post-motion structure recovery
Triangulation, block matching
6. State update
Kalman filters, bundle adjustment, etc.

Figure 4.1: Stages of a generic VO implementation annotated by related topics. An appearance-based method is a special case where Stages 2 and 4 are combined in an iterative egomotion estimation loop, skipping Stage 3.

frame $j-1$ to j, and concatenating the solved local transforms to find a global one:

$$\mathbf{T}_j = \Delta\mathbf{T}_j\mathbf{T}_{j-1}\,, \tag{4.1}$$

with the first frame being set as the reference, i.e. $\mathbf{T}_1 = \mathbf{I}$.

Stages of a typical visual odometer are shown in Fig. 4.1. The flow starts with some pre-processing tasks such as image rectification and feature detection as a new frame arrives. The identified key points are then associated with elements in the previously tracked point set to establish image correspondences which are then used at the egomotion estimation stage to solve for the camera's local transform.

Once solved, the egomotion is used to refine scene structure, and the new measures are integrated into the current state of the system (the integration is also known as *filtering*). The VO process, considered in this chapter, is seen as a front-end of a SLAM system which has some back-end stages that deal with long-term tracking and filtering as introduced in Chapter 5.

Based on how egomotion is solved, VO implementations are categorised as being either feature-based or appearance-based methods. The methods have their own strengths and weaknesses, as well as applicable types of sensor configurations. In this section we review these perspectives, preparing for the formulation of a unifying framework in the next section.

4.1.1 Feature-based methods

Feature-based methods detect image features and extract their descriptors for establishing sparse correspondences between frames, following the approach previously described in Section 3.3.2. Because a feature transform, in general, does not preserve image geometry, it is difficult to enforce the epipolar constraint in the feature space. However, if doing so, a robust outlier rejection strategy is required to remove incorrect point correspondences before motion can be reliably estimated.

The *random sample consensus* (RANSAC) technique, proposed by Fischler et al. (Fischler & Bolles, 1981), is nowadays considered to be a standard outlier-rejection paradigm in the context of SfM and VO (Kitt, Geiger & Lategahn, 2010; Fraundorfer & Scaramuzza, 2012).

Assume that $\mathcal{F}$ and $\mathcal{F}'$ are features in two frames, and $\mathcal{M} = \{\chi_i \leftrightarrow \chi_i'\}$, where $\chi_i \in \mathcal{F}$ and $\chi_i' \in \mathcal{F}'$ is a mapping established by descriptor matching between two feature sets.

The RANSAC technique first randomly selects a subset of N_{SAMPLE} elements from $\mathcal{M}$. A best-fit model hypothesis $\hat{f}$ is then generated from the samples. The hypothesis is applied to the population $\mathcal{M}$ to establish the consensus set:

$$\mathcal{C} = \{\chi_i \leftrightarrow \chi_i' \in \mathcal{M} \mid \delta(\chi_i, \chi_i'; \hat{f}) \leq \varepsilon\} \tag{4.2}$$

where δ measures the fitness of the model, and ε is a threshold for accepting a feature match as an inlier.

The hypothesis $\hat{f}$ is said to fit the data if $|\mathcal{C}| \ / \ |\mathcal{M}| \geq \alpha$ where $\alpha \in (0, 1]$ is a predefined threshold. Such a process is repeated for N_{TRIAL} times, and the hypothesis f that maximises the size of the consensus set is elected as a true model. The matches that do not fit the model well are treated as outliers.

A typical choice in VO is to use an essential matrix as $\hat{f}$, an epipolar distance function as δ, and a five-point solver (Nistér, 2004) for finding the best-fit model, given $N_{\text{SAMPLE}} = 5$ sample correspondences (see Section 4.3.1 for details).

As the RANSAC technique is a non-deterministic iterative approach, the filtering can be time consuming when the best-fit estimation of $\hat{f}$ is computational expensive. A heuristic strategy is often deployed to decide an upper bound to stop iterating earlier to obtain a statistically accurate estimate at a desired confidence.

Assuming we have confidence that $\mathcal{M}$ contains at least α percent inliers, then the probability that at least one element in sample set $S \subseteq \mathcal{M}$ is an outlier equals $1 - \alpha^{|S|}$ given that the samples are drawn independently from $\mathcal{M}$.

The probability that the sample set always contains at least one outlier after n iterations is then $(1 - \alpha^{|S|})^n$. Let $0 < p < 1$ be the certainty that a valid model f is found. It follows that $1 - p = (1 - \alpha^{|S|})^n$. The upper bound of parameter N_{TRIAL} can therefore be estimated by

$$n = \frac{\log(1 - p)}{\log(1 - \alpha^{|S|})} \tag{4.3}$$

where $|S| = N_{\text{SAMPLE}}$.

A number of improved RANSAC methods has been proposed such as *M-estimator sample and consensus* (MSAC) (Torr & Zisserman, 2000) or *preemptive RANSAC* (Nistér, 2003). For a comprehensive comparison on RANSAC-type methods, readers are referred to the study by Choi et al. (S. Choi, Kim & Yu, 2009).

Filtered image correspondences $\mathcal{C} \subseteq \mathcal{M}$ are used to estimate the camera's egomotion,

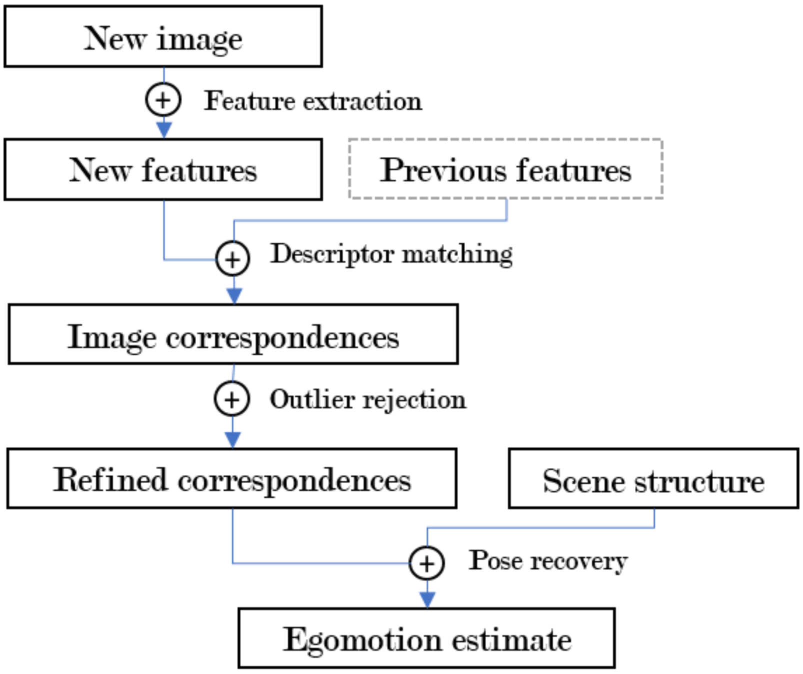


Figure 4.2: Egomotion estimation of a feature-based VO implementation

applying one of the approaches covered in Section 3.4. The process is depicted by Fig. 4.2. An advantage of using a feature transform is to obtain a set of accurate point correspondences even with a large image displacement, under which circumstances an optical flow algorithm usually would fail. A comparison on the influence of different image features on egomotion estimation is provided in Appendix B.

4.1.2 Appearance-based methods

Appearance-based VO methods do not use any pre-established image correspondences to find the camera's egomotion. The correspondences are instead associated on-the-fly by perspective warping using pixel depths and a motion hypothesis while egomotion is being solved.

As depicted by Fig. 4.3, egomotion estimation is achieved by iteratively minimising the photometric difference between warped pixels and a target image. Details of the photometric aligning process are described later in Section 4.3.3.

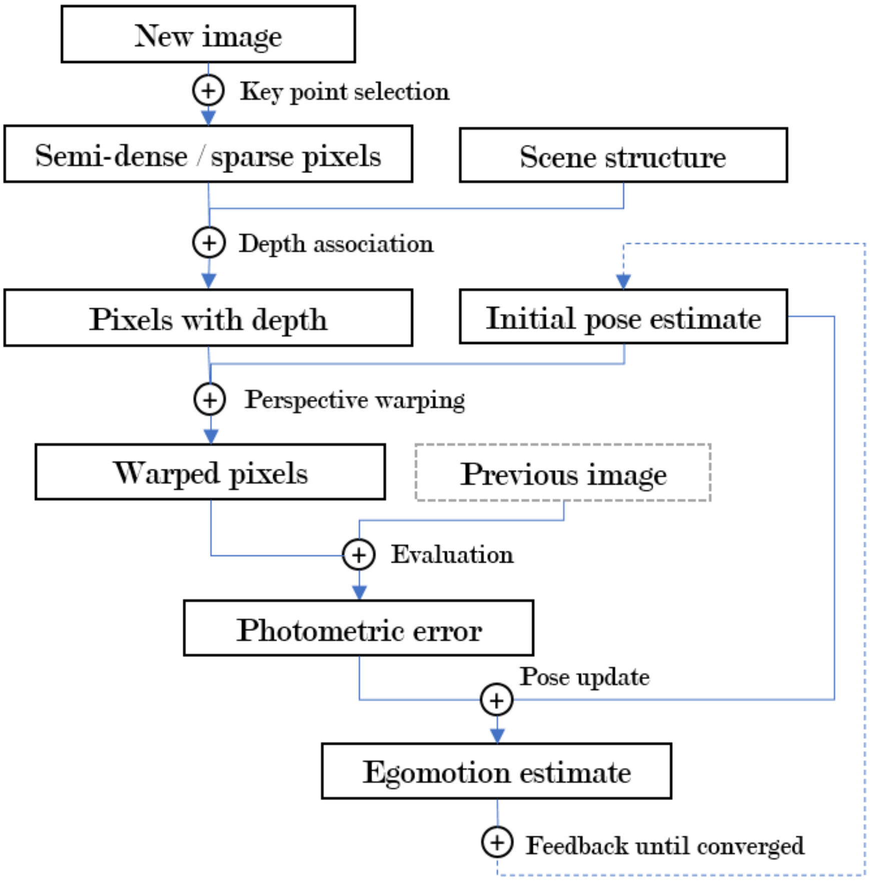


Figure 4.3: Egomotion estimation of a direct VO implementation. The egomotion is iteratively estimated through a nonlinear photometric error minimisation process.

Direct alignment in the image domain is well-preserving the epipolar condition. Furthermore, warping sparse or semi-dense points is much more computationally efficient than performing a feature transform with descriptor matching in a high-dimensional space. However, outliers may exist due to the following:

- A pixel is wrongly warped due to inaccurate depth.
- A warped pixel belongs to a moving object.
- A warped pixel is occluded.
- Pixel intensity changes due to varied illumination, sensor exposure, or surface specularity.

A feature-based method comes with robustness against these situations, due to the outlier rejection stage. In the next section we provide a unifying egomotion-estimation framework based on strengths of feature- and appearance-based methods.

4.2 A unifying framework

Egomotion estimation by a conventional VO system is based on a single objective: direct methods use a photoconsistency objective that aims at minimising a photometric alignment error while feature-based methods often target the minimisation of a re-projection error following a perspective alignment model.

An outlier rejection technique may take into account a secondary model such as epipolar alignments; nevertheless, it is not tightly integrated into the egomotion estimation stage. Mono-objective implementations can be less reliable when collected data is severely corrupted to fit the target model.

Based on the observed shortcoming, we propose a unifying framework that achieves egomotion estimate targeting multiple objectives, as shown in Fig. 4.4. The integration of multiple models improves a system's robustness to work in a complex scene such as a street.

Our unifying framework takes advantages from both categories of existing VO methods. In the feature-based category, a set of image features is identified and matched in feature space to establish initial correspondences $\mathcal{M}$. A subset of the correspondences are drawn to provide a motion hypothesis by a pose solver. The hypothesis is then verified by a set of N models built from a variety of criteria which may include epipolar constraints, perspective alignments, or photoconsistencies.

The verification results in a set of consensus sets $(\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_N)$, and the hypothesis are elected as being the candidate estimate if the size of the union consensus set is significant and larger than the consensus set of the current candidate.

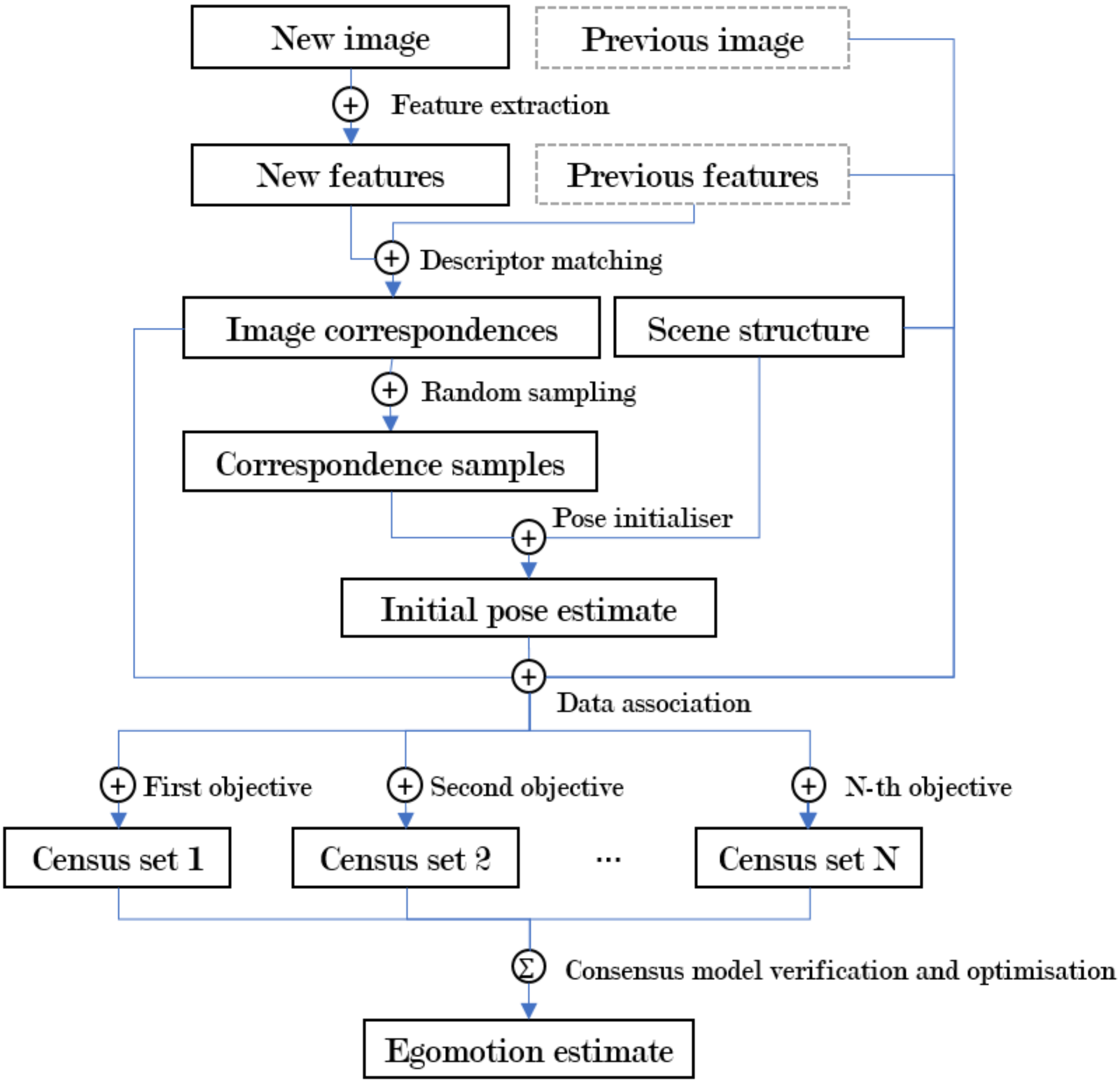


Figure 4.4: Egomotion estimation using a unifying multi-objective VO implementation. A set of models is selected to build objectives from the data aggregated from matched features, images, as well as from scene depth.

A modified RANSAC process, following the pseudo-code in Fig. 4.5, works more like a direct VO approach which verifies a motion hypothesis using dynamically associated correspondences based on it through the egomotion estimation stage. In Section 4.3 we review a number of alignment models that can be adopted for verification of a motion hypothesis.

A nonlinear refinement process is carried out to further optimise the elected motion candidate after a RANSAC process. The optimisation is subject to the models and their

```
Input  : Image correspondences $\mathcal{M}$, pose solver $h$, set of objective models $\varphi$,
         number of iterations $N_{\text{TRIAL}}$
Output : Egomotion estimate $\mathbf{T}$
$\mathbf{T} \leftarrow \mathbf{I}$;
$n \leftarrow 0$;
for $k \leftarrow 1$ to $N_{TRIAL}$ do
    $S_k \leftarrow$ DrawSamples($\mathcal{M}$);
    $\mathbf{T_k} \leftarrow$ SolvePose($h, S_k$);
    $n_k \leftarrow 0$;
    for $i \leftarrow 1$ to $|\Phi|$ do
        $\mathcal{C}_i$ = FindInliers($\varphi_i$, $\mathbf{T_k}$);
        $n_k \leftarrow n_k + |\mathcal{C}_i|$;
    end
    if $n_k > n$ then
        $\mathbf{T} \leftarrow \mathbf{T}_k$;
        $n \leftarrow n_k$;
    end
end
```

Figure 4.5: RANSAC process to find consensus egomotion among multiple objectives.

consensus sets $\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_N$ to avoid biased adjustments towards a single target. The multi-objective optimisation process is detailed in Section 4.4.

4.3 Alignment models

In the context of VO and SLAM there are a number of measures or metrics adopted to evaluate an estimated camera pose $[\mathbf{R} \;\; \mathbf{t}]$. In this section we review a number of alignment models as well as how uncertainty of a measurement can be taken into account to build a model.

Uncertainty estimation in this context is necessary to normalise the evaluated quantities to support the integration of different objective functions in a statistically meaningful way, as discussed in Section 4.4. In this work we follow a Gaussian framework and apply the Mahalanobis metric to achieve MLE estimation.

4.3.1 Epipolar alignment

Given an image point $\mathbf{x} = (x, y)^\top$ in the current frame and $(\mathbf{R}, \mathbf{t})$ for the motion of the camera, the corresponding epipolar line can be obtained in the next frame identifying the search domain for the corresponding image point $\mathbf{x}' = (x', y')^\top$. Such a 2D-to-2D point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ is useful for evaluating the correctness of a motion hypothesis.

As back-projected rays through $\mathbf{x}$ and $\mathbf{x}'$ have to be co-planar, this leads to the well-known epipolar constraint (Hartley & Zisserman, 2004; Klette, 2014)

$$\tilde{\mathbf{x}}'^\top \mathbf{K}^{-\top} [\mathbf{t}]_\times \mathbf{R} \mathbf{K}^{-1} \tilde{\mathbf{x}} = 0 \tag{4.4}$$

where $[\mathbf{t}]_\times$ is the skew-symmetric matrix form of vector $\mathbf{t}$, $\mathbf{K}$ is the camera matrix, and $\tilde{\mathbf{x}} = (x, y, 1)^\top$ are homogeneous coordinates of an image point in vector form.

In practice, the ideal equality of Eq. (4.4) is never true as a result of numerical computations, errors in correspondences, or inaccuracy of the motion hypothesis. For the last factor, from a set of correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$, one may obtain the residual terms

$$\varphi(\mathbf{x}, \mathbf{x}'; \mathbf{R}, \mathbf{t}) = \tilde{\mathbf{x}}'^\top \mathbf{F} \tilde{\mathbf{x}} \tag{4.5}$$

where $\mathbf{F} = \mathbf{K}^{-\top} [\mathbf{t}]_\times \mathbf{R} \mathbf{K}^{-1}$ is the fundamental matrix encoding the given epipolar geometry. Algebraic distances, however, are biased as image points far away from the epipole tend to be over-penalised.

A geometrically meaningful modelling approach is to measure the shortest distance between $\mathbf{x}'$ and the corresponding epipolar line $\mathbf{l} = \mathbf{F}\tilde{\mathbf{x}} = (l_0, l_1, l_2)^\top$. The distance is given by

$$\delta(\mathbf{x}', \mathbf{l}) = \frac{|\tilde{\mathbf{x}}'^\top \mathbf{F} \tilde{\mathbf{x}}|}{\sqrt{l_0^2 + l_1^2}} . \tag{4.6}$$

As the observation $\mathbf{x}'$ also introduces an epipolar constraint on $\mathbf{x}$, we have that

$$\delta(\mathbf{x}, \mathbf{l}') = \frac{|\tilde{\mathbf{x}}'^{\top} \mathbf{F} \tilde{\mathbf{x}}|}{\sqrt{l'^2_0 + l'^2_1}} \tag{4.7}$$

where $\mathbf{l}' = \mathbf{F}^{\top} \tilde{\mathbf{x}}'$ denotes the epipolar line in the first view. By applying symmetric measurements on the point-to-epipolar-line distances, the energy function defined by Eq. (4.5) is now revised as follows:

$$\varphi(\mathbf{x}_i, \mathbf{x}'_i; \mathbf{R}, \mathbf{t}) = \delta^2(\mathbf{x}'_i, \mathbf{F}\mathbf{x}_i) + \delta^2(\mathbf{x}_i, \mathbf{F}^{\top}\mathbf{x}_i) \tag{4.8}$$

This yields geometric errors in pixel locations.

A noise-tolerant variant is to treat the correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ as a deviation from the ground truth $\hat{\mathbf{x}} \leftrightarrow \hat{\mathbf{x}}'$. When the differences $\|\mathbf{x} - \hat{\mathbf{x}}\|$ and $\|\mathbf{x}' - \hat{\mathbf{x}}'\|$ are believed to be small, the sum of squared mutual geometric distances can be approximated by

$$\delta^2(\hat{\mathbf{x}}, \hat{\mathbf{l}}') + \delta^2(\hat{\mathbf{x}}', \hat{\mathbf{l}}) \approx \frac{(\tilde{\mathbf{x}}'^{\top} \mathbf{F} \tilde{\mathbf{x}})^2}{l^2_0 + l^2_1 + l'^2_0 + l'^2_1} . \tag{4.9}$$

This first-order approximation to the geometric error is known as the *Sampson distance* (Hartley & Zisserman, 2004).

As the computation of these epipolar errors only uses 2D correspondences, the energy model can be useful when 3D structures of a scene are not known (i.e. in case of monocular VO). The five-point method, credited to Nistér (Nistér, 2004), may be applied (as discussed before). The absolute scale of $\mathbf{t}$, however, is not recoverable without any reference in the 3D space, as explained in Section 3.4.2.

4.3.2 Projective alignment

A scale-aware alignment model uses not only 2D projections but also 3D data. The 3D-to-2D correspondences are constrained by a sensor's projection model as well as the egomotion of the system. The *reprojection error* (RPE) is defined as the geodesic distance between the projection of a 3D point and its predicted position in an image. This error is considered to be the "gold standard" in SfM and VO (Hartley & Zisserman, 2004; Engels, Stewénius & Nistér, 2006). In particular, the residual is defined by

$$\varphi(\mathbf{g}, \mathbf{x}'; \mathbf{R}, \mathbf{t}) = \|\mathbf{x}' - \pi\left(\mathbf{R}\mathbf{g} + \mathbf{t}\right)\|_{\mathbf{\Sigma}}^{2} \tag{4.10}$$

where $\mathbf{g} = (X, Y, Z)^{\top}$ is the current 3D location of a feature, $\mathbf{x}' = (x, y)^{\top}$ are the feature's image coordinates in the next frame, $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ is the projection function that maps a 3D point into the 2D image coordinates, and $\mathbf{\Sigma}$ is the 2×2 error covariance matrix of the correspondence.

When a backward correspondence $\mathbf{g}' \leftrightarrow \mathbf{x}$ is known, Eq. (4.10) can be modelled in an inverse mapping form as follows:

$$\varphi(\mathbf{g}', \mathbf{x}; \mathbf{R}, \mathbf{t}) = \|\mathbf{x} - \pi\left(\mathbf{R}^{\top}\left(\mathbf{g}' - \mathbf{t}\right)\right)\|_{\mathbf{\Sigma}}^{2} . \tag{4.11}$$

When an error covariance matrix is associated to the 3D point g, it has to be properly propagated to the image and integrated into the covariance matrix equipped with $\mathbf{x}'$. In particular, the covariance used in Eq. (4.10) is replaced by the integrated covariance matrix $\bar{\mathbf{\Sigma}}$:

$$\bar{\mathbf{\Sigma}} = \mathbf{J}_{\pi}\mathbf{R}\mathbf{\Sigma}\mathbf{R}^{\top}\mathbf{J}_{\pi}^{\top} + \mathbf{\Sigma}' \tag{4.12}$$

where $\mathbf{J}_{\pi}$ is the 2×3 Jacobian of π at $\mathbf{R}\mathbf{g} + \mathbf{t}$, $\mathbf{\Sigma}$ is the 3×3 covariance matrix of g, and $\mathbf{\Sigma}'$ is the 2×2 covariance matrix of $\mathbf{x}'$.

The geodesic reprojection error has been a widely adopted energy model in camera re-sectioning, including calibration, pose estimation, or bundleadjustment (Engels et al., 2006). Its closed-form linear solution has been extensively studied in the domain of PnP problems (Gao, Hou, Tang & Cheng, 2003).

A popular solver is the efficient algorithm due to Lepetit et al. (Lepetit et al., 2009). Section 3.4.1 provides details of this linear solver. A linear solution is usually iteratively refined using a derivative-based minimiser (e.g. a Gauss-Newton algorithm). It has been shown that the projective alignment can be further regularised using the aforementioned epipolar alignment to reduce the impact of noisy 3D measurements (Chien & Klette, 2017).

4.3.3 Photometric alignment

The photometric criterion is based on comparisons between raw or transformed image intensities. Direct methods make direct alignments on pixel intensities while feature-based methods rely on descriptor matching after a feature transform.

If an image correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ is not observable, one may perform direct photometric matching using a motion hypothesis as shown in Section 3.4.3. In this case, a matching residual is defined over intensity images I and I' as follows:

$$\varphi(\mathbf{x}, \mathbf{g}; \mathbf{R}, \mathbf{t}) = \| I(\mathbf{x}) - I'[\pi(\mathbf{R}\mathbf{g} + \mathbf{t})] \|^2 \tag{4.13}$$

assuming that the 3D coordinates $\mathbf{g} = (X, Y, Z)$ of image point $\mathbf{x} = (x, y)$ are known. This is also known as *correspondence-free egomotion estimation*, and it is pervasively used by direct VO techniques.

As discussed previously, Eq. (4.13) can be extended to use a block of pixels instead of single pixel intensities, given a properly defined metric such as SAD or SSD.

The evaluation of Eq. (4.13) is computationally expensive compared to all the other geodesic criteria considered in this work. It invokes a rigid transform, a perspective transform, and two image sub-sampling procedures. As the minimum of photometric errors can only be approached iteratively, such an expensive cost function will need to be invoked repeatedly for constructing numerically the Jacobian matrix.

To ease the incurred burden, various direct methods adopt an inverse-compositional form of the residual term (Forster et al., 2014)

$$\varphi^k(\mathbf{g}, \xi_k; \Delta\xi) = \|I\Big[\pi\Big(\Delta\mathbf{T} \cdot \mathbf{g}\Big)\Big] - I'\Big[\pi\Big(\mathbf{T}_k(\xi_k) \cdot \mathbf{g}\Big)\Big]\|^2 \tag{4.14}$$

where we represent $\mathbf{R}$ and $\mathbf{t}$ by the twist ξ such that $\Delta\mathbf{T} = \exp(\Delta\xi)$ and $\mathbf{T}_k = \exp(\xi_k)$. Along with the inverse form of

$$\xi_{k+1} = \Delta\xi_k^{-1} \circ \xi_k \tag{4.15}$$

the Jacobian of ϕ can be written in the chain form

$$\frac{\partial\varphi}{\partial\Delta\xi}(\mathbf{g}, \xi_k) = \frac{\partial I}{\partial x}\Big|_{x=\mathbf{x}} \cdot \frac{\partial\pi}{\partial y}\Big|_{y=\mathbf{g}} \cdot \frac{\partial\mathbf{T}}{\partial\xi}\Big|_{\xi=\mathbf{0}} \cdot \mathbf{g} \tag{4.16}$$

which is independent of the current motion hypothesis ξ_k.

The first term of Eq. (4.16) is the gradient of base image I at key point $\mathbf{x}$ which requires only one-time evaluation at the beginning of the minimisation procedure; the second and third term can be calculated symbolically, and the last term is constant, for each tracked key point.

When the covariance of g is known, it can be taken into account to normalise Eq. (4.13), following

$$\varphi(\mathbf{x}, \mathbf{g}; \mathbf{R}, \mathbf{t}) = \|I(\mathbf{x}) - I'[\pi(\mathbf{R}\mathbf{g} + \mathbf{t})]\|^2_{\mathbf{\Sigma}} \tag{4.17}$$

where $\mathbf{\Sigma}$ is the propagated covariance

$$\bar{\mathbf{\Sigma}} = \mathbf{J}_I \mathbf{J}_\pi \mathbf{R} \mathbf{\Sigma} \mathbf{R}^\top \mathbf{J}_\pi^\top \mathbf{J}_I^\top \tag{4.18}$$

with $\mathbf{J}_I = \left[\frac{\partial I}{\partial x}(\mathbf{x}') \;\; \frac{\partial I}{\partial y}(\mathbf{x}') \right]$ the gradient of the image manifold at $\mathbf{x}' = \pi(\mathbf{R}\mathbf{g} + \mathbf{t})$.

4.3.4 Rigid alignment

If a dense depth map is available and the establishment of 3D point correspondences is straightforward, then a rigid alignment can also be used to measure the fitness of a motion hypothesis.

Given 3D-to-3D correspondences $\mathbf{g} \leftrightarrow \mathbf{g}'$, where $\mathbf{g} = (X, Y, Z)^\top$ and $\mathbf{g}' = (X', Y', Z')^\top$, the energy model is defined by

$$\varphi(\mathbf{g}, \mathbf{g}'; \mathbf{R}, \mathbf{t}) = \| \mathbf{g}' - (\mathbf{R}\mathbf{g} + \mathbf{t}) \|_{\mathbf{\Sigma}}^2 \tag{4.19}$$

where $\mathbf{\Sigma}$ denotes the 3×3 error covariance matrix.

The formulation can be ill-behaved for far points if 3D coordinates are derived from a disparity map, due to the non-linearity of disparity-to-depth conversion. It is therefore critical to model the covariance matrix properly.

If 3D coordinates are obtained by using a two-view triangulation function $\tau : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^3$, then the covariance matrix can be modelled as

$$\mathbf{\Sigma} = \mathbf{J}_\tau \begin{pmatrix} \mathbf{\Sigma}_{\mathbf{x}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{\mathbf{x}'} \end{pmatrix} \mathbf{J}_\tau^\top \tag{4.20}$$

where $\mathbf{\Sigma}_{\mathbf{x}}$ and $\mathbf{\Sigma}_{\mathbf{x}'}$ are the 2×2 error covariance matrices of image points $\mathbf{x} = (x, y)^\top$

and $\mathbf{x}' = (x', y')^\top$, respectively, and $\mathbf{J}_\tau$ is the 3×4 Jacobian matrix

$$\mathbf{J}_\tau = \left[\frac{\partial \tau}{\partial x}(\mathbf{x}, \mathbf{x}') \;\; \frac{\partial \tau}{\partial y}(\mathbf{x}, \mathbf{x}') \;\; \frac{\partial \tau}{\partial x'}(\mathbf{x}, \mathbf{x}') \;\; \frac{\partial \tau}{\partial y'}(\mathbf{x}, \mathbf{x}') \right] \tag{4.21}$$

with respect to correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$, used to triangulate $\mathbf{g} = (X, Y, Z)^\top$ (Maimone et al., 2007).

The rigid model has closed-form solutions that are guaranteed to minimise Eq. (4.19). A popular choice is based on quaternion parametrization and SVD, as shown by Horn (Horn, 1987).

4.4 Egomotion estimation

Based on the selected alignment models, a set of mappings is built and input to the multi-objective RANSAC process described in Section 4.2. For each correspondence $\chi \leftrightarrow \chi' \in \mathcal{M}$, the building of model data is as follows:

- Let $\rho_j(\chi)$ be the image coordinates of feature χ in the j-th frame; the 2D-to-2D correspondences $\rho_{j-1}(\chi) \leftrightarrow \rho_j(\chi')$ are used to build the epipolar constraints, denoted by $\mathcal{M}_{\text{EPI}}$.

- Let $\bar{g}_j(\chi)$ be the estimated 3D coordinates of feature χ in the j-th frame; the correspondences $\bar{g}_{j-1}(\chi) \leftrightarrow \rho_k(\chi')$ are used to build the projection constraints, denoted by $\mathcal{M}_{\overrightarrow{\text{RPE}}}$. We also make use of constraints $g_j(\chi') \leftrightarrow \rho_{j-1}(\chi)$ to build backward reprojection constraints $\mathcal{M}_{\overleftarrow{\text{RPE}}}$ whenever available.

- The intensity-3D-intensity correspondences $I_{j-1}\big[\rho_{j-1}(\chi)\big] \leftrightarrow \bar{g}_{j-1}(\chi) \leftrightarrow I_j\big[\rho_j(\chi')\big]$ are used to instantiate a set of photometric constraints, denoted by $\mathcal{M}_{\text{PHOTO}}$.

- If the measure $g_j(\chi)$ of a feature's 3D coordinates in the new frame is available (either from a disparity map, a LiDAR scan, or any other sources), we construct a set of 3D-to-3D constraints $\hat{g}_{j-1}(\chi) \leftrightarrow g_j(\chi')$, and have it denoted by $\mathcal{M}_{\text{RIGID}}$.

All the constructed mappings $\mathcal{M}_{\text{EPI}}$, $\mathcal{M}_{\overrightarrow{\text{RPE}}}$, $\mathcal{M}_{\overleftarrow{\text{RPE}}}$, $\mathcal{M}_{\text{RIGID}}$, and $\mathcal{M}_{\text{PHOTO}}$ are collaboratively used in the introduced RANSAC process to remove outliers.

First, at the initialisation stage, a minimum set of correspondences is randomly withdrawn from one of the five mappings. From the samples, an initial motion hypothesis $\bar{\xi}$ is solved by the closed-form solver associated with the chosen mapping. At the verification stage, the hypothesis is applied to each class.

By means of the appropriate energy model, introduced in Section 4.3, the data terms are evaluated and the inliers are found as the correspondences achieving an error below a pre-defined class-specific threshold. If the population of inliers, summed over all the classes, achieves a record high, the hypothesis is taken as the best model. Such a process goes until a stopping criterion is met.

The closed-form solver and associated energy model for each class are summarised in Table 4.1. Note that the $\mathcal{M}_{\text{EPI}}$ class is excluded at the estimation stage, as the translation of estimated motion $\bar{\xi}$ does not have an absolute unit, prohibiting it from being used to evaluate data terms in other classes. The $\mathcal{M}_{\text{PHOTO}}$ class is also excluded because there is no closed-form solver for the photometric alignment problem.

The best-fit model $\bar{\xi}$ from the RANSAC process serves as the initial guess at the

Table 4.1: Summary of alignment model and data terms

Model	Mapping	Alignment	Type	Solver	N_{SAMPLE}
Φ_{EPI}	$\mathcal{M}_{\text{EPI}}$	Epipolar	2D-to-2D	Five-point [1]	5
Φ_{RPE}	$\mathcal{M}_{\overrightarrow{\text{RPE}}}$, $\mathcal{M}_{\overleftarrow{\text{RPE}}}$	Perspective	3D-to-2D	EPnP [2]	5
Φ_{RIGID}	$\mathcal{M}_{\text{RIGID}}$	Rigid	3D-to-3D	SVD [3]	4
Φ_{PHOTO}	$\mathcal{M}_{\text{PHOTO}}$	Photometric	Int.-to-int. [*]	N/A	N/A

[*] "Intensity-to-intensity".

[1] Five-point root-finding algorithm (Nistér, 2004)

[2] Efficient PnP solver (Lepetit et al., 2009)

[3] Quaternion-based SVD approach (Horn & Schunck, 1981)

non-linear optimisation stage over the integrated energy model

$$\Phi(\xi) = \Phi_{\text{RPE}}(\xi) + \Phi_{\text{EPI}}(\xi) + \Phi_{\text{RIGID}}(\xi) + \Phi_{\text{PHOTO}}(\xi) \tag{4.22}$$

with each sub-objective summarising the squared residuals instantiated from the census set of the corresponding class. Note that the combination does not use per-class weightings as the residuals are already normalised by the estimated error covariance, as discussed in Section 4.3. Function Φ is minimised using the iterative process described in Section 3.4.4.

4.5 Post-motion processing

After a motion hypothesis is solved, it can be used to refine a previously estimated state which (in this work) refers to the 3D structure of tracked features. It is also helpful in recovering features lost by descriptor matching or the outlier-rejection process.

4.5.1 Inlier injection

Those long-term features that have been successfully tracked over a period of time contribute to accurate egomotion estimation more than the features that survive only a few frames (Badino et al., 2013). Recovery of lost features not only increases the number of high-quality features but also reduces false instantiation of landmarks due to disrupted observations which is critical to eliminate redundant structure parameters at the global optimisation stage, to be discussed in the next chapter.

A feature $\chi \in \mathcal{F}$ is *lost* if it is not included in any consensus mapping $\mathcal{C}$ that contributed to the final egomotion estimate ξ.

A KLT tracking algorithm (Baker & Matthews, 2004) is applied on each lost feature in F to find its image position in the current frame, and an augmented feature set is created by gathering all the results. Two outlier rejection techniques are performed on the augmented set afterwards.

First, backward tracking is conducted to map augmented features in the current frame to the previous one; if the distance between a feature's originating location and the back-tracked pixel is larger than a tolerable distance of, say, 0.5 pixels, the feature will be removed from the augmented set. Such a check is known as *bidirectional consistency* (J. Choi, Kim, Oh & Kweon, 2014) test.

Second, this test is followed by enforcing the epipolar constraint based on ξ. A tracked feature is rejected if its updated position deviates from the epipolar line over a reasonable range which (in this research) is set to the same threshold used to enforce bidirectional consistency.

If a lost feature's depth is known in the previous frame, its 2D coordinates can also be predicted by projecting the 3D coordinates to the current image, given the egomotion has already been solved (Badino et al., 2013; Forster et al., 2014). In this case we also perform a scoped epipolar search following the manner described in Section 3.3.1. Figure 4.6 demonstrates examples of the introduced feature-recovery strategies.

4.5.2 Structure recovery and recursive filtering

Census mappings (from all the adopted alignment models plus the augmented mapping) form a feature flow which is used by a two-view triangulator to measure the 3D coordinates of the features. Integration is required if a triangulated feature already has a previously estimated depth, either from another triangulation, a disparity-depth conversion, or other sources. In this work, a Bayesian framework is adopted to actualise the integration by taking the certainty of each measurement into account.

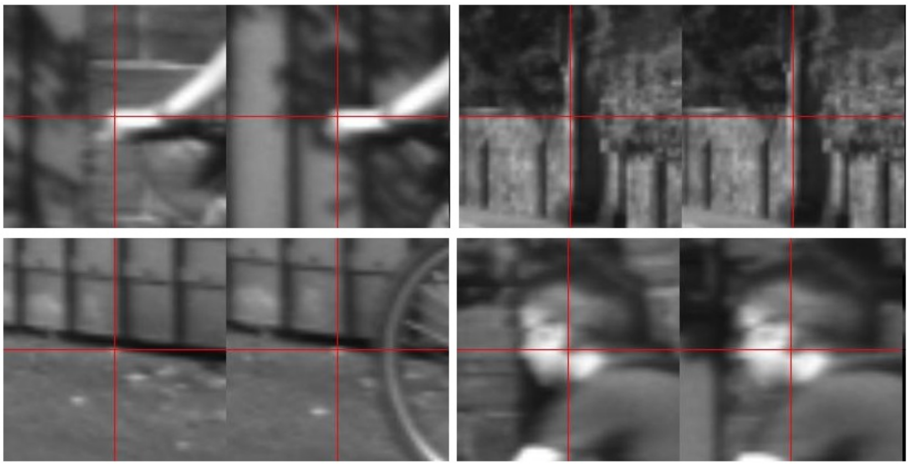


Figure 4.6: Recovery of lost features in sub-pixel accuracy. Four features lost through the VO pipeline have been successfully recovered using solved egomotion and their 3D coordinates. Each pair of two image windows shows on the left a feature in the previous frame and on the right its recovered position in the current frame. Please note that the correspondences are established without any descriptor matching.

In the univariate case, given two estimates (x_0, σ_0^2) and (x_1, σ_1^2), the merged estimates follow

$$\bar{x} = \frac{\sigma_1^2 x_0 + \sigma_0^2 x_1}{\sigma_0^2 + \sigma_1^2}, \quad \bar{\sigma}^2 = \frac{\sigma_0^2 \sigma_1^2}{\sigma_0^2 + \sigma_1^2} \quad . \tag{4.23}$$

In the related literature, variances are often modelled as inverse weightings (Badino et al., 2013). Using weightings $w_0 = 1/\sigma_0^2, w_1 = 1/\sigma_1^2$, the updating formula becomes

$$\bar{x} = \frac{w_0 x_0 + w_1 x_1}{w_0 + w_1}, \quad \bar{w} = w_0 + w_1 \quad . \tag{4.24}$$

An alternative form of Eq. (4.23) is

$$\bar{x} = x_0 + G \cdot (x_1 - x_0), \quad \bar{\sigma}^2 = G\sigma_1^2 \tag{4.25}$$

where the coefficient

$$G = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2} \tag{4.26}$$

is known as the *Kalman gain* (Civera et al., 2008).

Filtering can be generalised to the multivariate case. Let $(\mathbf{x}_0, \mathbf{\Sigma}_0)$ and $(\mathbf{x}_1, \mathbf{\Sigma}_1)$ be the estimates to be merged. The Kalman gain is computed in its matrix form

$$\mathbf{G} = \mathbf{\Sigma}_0(\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1)^{-1} \tag{4.27}$$

and the update is as follows:

$$\bar{\mathbf{x}} = \mathbf{x}_0 + \mathbf{G}(\mathbf{x}_1 - \mathbf{x}_0), \quad \bar{\mathbf{\Sigma}} = \mathbf{G}\mathbf{\Sigma}_1 \quad . \tag{4.28}$$

In the case of $\det(\mathbf{\Sigma}_0) \gg \det(\mathbf{\Sigma}_1)$, where the uncertainty of $\mathbf{x}_0$ is much higher than $\mathbf{x}_1$, it appears that $(\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1)^\top \approx \mathbf{\Sigma}_0^{-1}$ as the error covariance $\mathbf{\Sigma}_0$ is dominating the inverse over $\mathbf{\Sigma}_1$, and therefore $\mathbf{G} \approx \mathbf{I}$. In this case, the new estimate is closer to $\mathbf{x}_1$ as it is more trustworthy.

In the context of VO, we consider $\mathbf{x}_0$ to be previously integrated 3D coordinates of a feature, and $\mathbf{x}_1$ the newly triangulated estimate transformed to the reference frame using the solved egomotion. The covariance matrices are estimated as described in Section 3.2.1. To properly model the system, we also take into account the uncertainty of the estimated egomotion.

Let $\bar{\mathbf{g}}_{j-1}$ be a previously integrated 3D position of a feature, and $\mathbf{g}_j$ the new estimate in Frame j. The fusion follows

$$\bar{\mathbf{g}}_j = \bar{\mathbf{g}}_{j-1} + \bar{\mathbf{\Sigma}}_{j-1}\left(\bar{\mathbf{\Sigma}}_{j-1} + \mathbf{\Sigma}_j\right)^{-1}\left[\mathbf{R}^\top\left(\mathbf{g}_j - \mathbf{t}\right) - \bar{\mathbf{g}}_{j-1}\right] \tag{4.29}$$

where covariance $\mathbf{\Sigma}_j$ is propagated by the solved egomotion $[\mathbf{R} \;\; \mathbf{t}]$, following

$$\mathbf{\Sigma}_j = \mathbf{R}^\top\left(\mathbf{J}_{\mathbf{x}\mathbf{x}'}\mathbf{\Sigma}_{\mathbf{x}\mathbf{x}'}\mathbf{J}_{\mathbf{x}\mathbf{x}'}^\top + \mathbf{J}_\xi\mathbf{H}^{-1}\mathbf{J}_\xi^\top\right)\mathbf{R} \tag{4.30}$$

via $\mathbf{J}_{\mathbf{x}\mathbf{x}'}$, the triangulator's 3×4 Jacobian with respect to image correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$

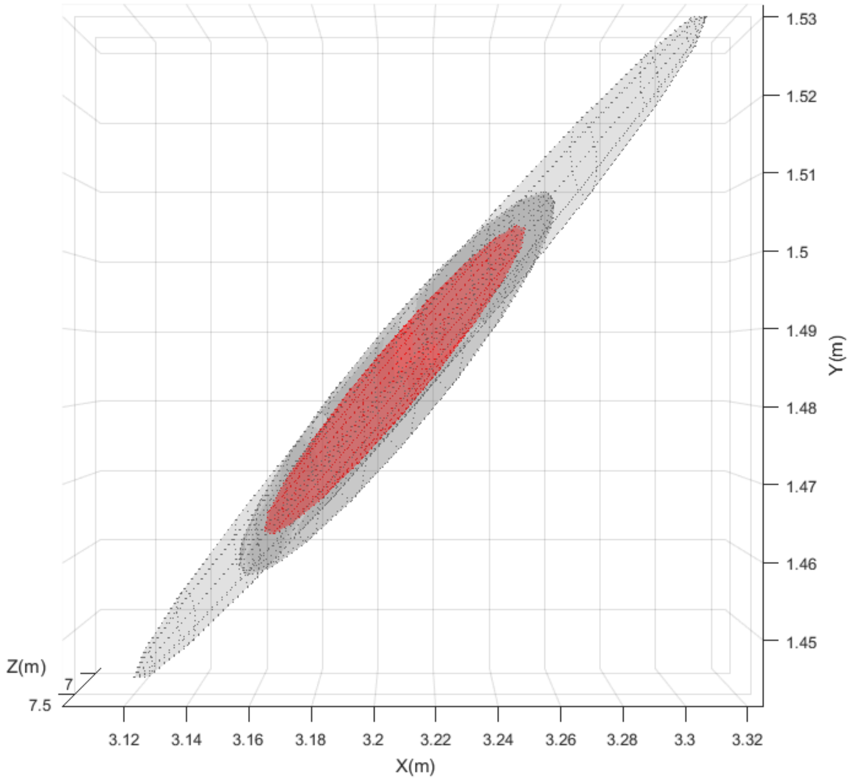


Figure 4.7: Two measures of a point at $Z = 7.33 \pm 0.54$m are fused. The outer error ellipsoid visualises the first measure which is based on disparity-depth conversion (see Section 3.2.2). The inner grey ellipsoid shows the second measure using a two-view, mid-point triangulation (see Section 3.2.1) based on solved egomotion. The ellipsoid showing the new estimate is shaded in red. A Bayesian filter tends to move the fusion towards the second estimate as it poses a less dispersed uncertainty due to a longer baseline.

of error covariance $\mathbf{\Sigma}_{\mathbf{xx}'}$ used to do the triangulation, $\mathbf{J}_\xi$ the 3×6 Jacobian with respect to egomotion estimation ξ, and $\mathbf{H}$ the 6×6 Hessian matrix obtained when ξ is solved by the Levenberg-Marquardt method (see Eq. (3.49) in Section 3.4.4).

Figure. 4.7 shows the fusion of two triangulations of the same point following the model. The first triangulation is based on a depth-disparity conversion following a stereo matching process by the time the exemplary feature is initially discovered. The second triangulation uses a recovered egomotion after the camera moves to a new position. Based on the established image correspondence, the measure finds the closest point of two back-projected rays. See Section 3.2 for the triangulation processes.

4.6 Experiments

4.6.1 Multi-objective egomotion estimation

We selected sequence `2011_09_26_drive_0027` from the `Road` category of the KITTI benchmark suite (Geiger et al., 2013) for evaluating the unified VO framework. The vehicle had travelled 350 metres (while recording 188 frames) on a single two-lane carriageway, with trucks and cars coming from the opposite lane.

The VO implementation computes disparity maps using a semi-global block matcher implemented in `OpenCV 3.3.0` ("The OpenCV Reference Manual: Camera Calibration and 3D Reconstruction", 2017). For each frame, SURF features (Bay, Tuytelaars & Van Gool, 2006) are detected and a 256-byte descriptor is extracted for each detected feature. The features are tracked through the image sequence. The images from the right camera are only used for disparity generation. The depth of a tracked feature is continuously integrated using a recursive Bayesian filter. No additional optimisation technique (e.g. bundle adjustment, lost feature recovery, or similar) has been deployed.

To test how each energy model affects the VO process, we tried all the possible combinations by enabling a subset of data terms among $\mathcal{M}_{\text{EPI}}$, $\mathcal{M}_{\overleftarrow{\text{RPE}}}$, $\mathcal{M}_{\text{RIGID}}$ and $\mathcal{M}_{\text{PHOTO}}$, for each test.

Note that the forward-projection term $\mathcal{M}_{\overrightarrow{\text{RPE}}}$ is always used as it is required to properly bootstrap the RANSAC process. This results in sixteen configurations. Due to the randomness introduced into the outlier-rejection stage, we carried out five trials for each configuration. Table 4.2 summarises motion drifts from eighty estimated trajectories.

In most cases, using additional energy model(s) significantly reduces the drift in estimated egomotion. Exceptions are observed in the cases of `xxRx`, `Bxxx`, and `BxRx`.

When the rigid alignment is solely imposed (`xxRx`), the drift only slightly reduces

Table 4.2: Egomotion estimation drifts (%) of different energy model combinations. Maximum and minimum value in each column are in bold.

Model [*]	Best	Worst	Mean	Std.	Model	Best	Worst	Mean	Std.
xxxx	4.97	5.54	5.21	0.27	Bxxx	**5.14**	5.99	5.41	0.34
xPxx	2.26	2.76	2.52	0.21	BPxx	1.99	2.50	2.23	0.21
xxRx	4.65	5.09	4.88	0.15	BxRx	5.10	**6.00**	**5.58**	**0.37**
xPRx	**1.84**	2.39	2.18	0.26	BPRx	1.96	2.56	**2.16**	0.26
xxxE	2.27	2.31	2.28	**0.01**	BxxE	2.21	**2.29**	2.24	0.03
xPxE	2.24	2.71	2.47	0.17	BPxE	2.17	2.48	2.31	0.11
xxRE	2.29	2.38	2.34	0.03	BxRE	2.18	2.31	2.24	0.05
xPRE	2.41	2.59	2.50	0.08	BPRE	2.21	2.40	2.33	0.08

[*] Letters B, P, R, and E, respectively, indicate the use of backward projection ($\mathcal{M}_{\overleftarrow{\text{RPE}}}$), photometric ($\mathcal{M}_{\text{PHOTO}}$), rigid ($\mathcal{M}_{\text{RIGID}}$), and epipolar ($\mathcal{M}_{\text{EPI}}$) alignment models.

by 0.35%. A result, worse than the forward-projection-only baseline configuration (xxxx), is found in BxRx where it is simultaneously applied with the back-projection alignment model. A similar result is observed when the backward projection is used (Bxxx). The loss of accuracy is due to the use of feature depths obtained in Frame $t + 1$, where the recursive Bayesian filter has not been applied, as the egomotion from t to $t + 1$ is not yet estimated (see Fig. 4.1).

Interestingly, the results show that, when the epipolar term is used with the forward projection model (xxxE), the VO process yields highly robust estimates with a very low standard deviation (0.01%). Such finding corresponds to work reported in (Chien & Klette, 2017).

We further compared the baseline model with the best case, worst case, and the case using all energy models. Accumulated drifts are plotted in Fig. 4.8. In the best case the accuracy is improved by 58.3%, while by imposing all four energy models, this option achieves an improvement by 54.8%.

We also profiled the run-time of each test case. The processing time for each frame are 230 ms and 200 ms for the best case and the baseline implementation, respectively. This time measurement indicates that the introduction of additional energy terms only

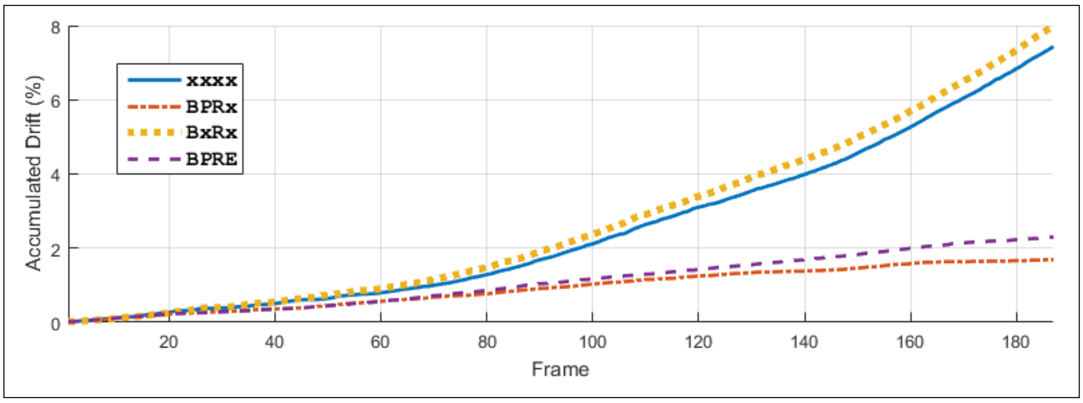


Figure 4.8: Drift analysis of the best (BPRx), worst (BxRx), all-enabled (BPRE), and the baseline model (xxxx)

incurs 13% more computational cost, while an improvement of above 50% in terms of accuracy is attainable.

4.6.2 Tracking topology

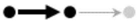
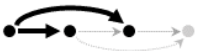
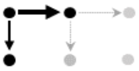
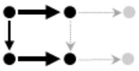
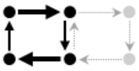
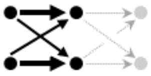
Conventionally, feature tracking is performed over every two consecutive frames, which forms a linear topology[1] in the pose graph. Few recent work suggests to deploy tracking in some more general topologies to improve the system's performance and robustness (Geiger, Ziegler & Stiller, 2011; Ci & Huang, 2016). For example, a circular matching pattern that tracks left image's features from Frame j to Frame $j+1$, crossing to the right image, back to Frame j, and finally back to the left image, was proved to be an effective approach to remove a false matching.

In this section we test the proposed VO framework in a variety of topological configurations, as summarised by Table 4.3.

The tracking topologies listed in the table are tested using the sequence chosen in Section 4.6.1. The detected SURF features are associated with depth data and tracked over left and optionally right image sequences. The KLT point tracker for lost feature

[1] "Topology" not in the sense of mathematical topology; here this term characterizes the connectivity between nodes of a graph.

Table 4.3: Feature tracking topologies.

Topology	Pattern	Description
	Linear	In the simplest configuration of VO, matching takes place from Frame j to $j + 1$.
	Look-ahead	Frame $j + 2$ is also tracked from j to avoid a feature drop due to single tracking failure. The topology has been deployed also for the purpose of outlier rejection using the trifocal tensor introduced by 3-view geometry (Song, Chandraker & Guest, 2013).
	Stereo linear	This is perhaps the most straightforward way for extending the feature tracking framework also to the right camera. It has been adopted by a number of common stereo VOs where left-right feature matching is performed for triangulating features' 3D coordinates (Olson, Matthies, Schoppers & Maimone, 2000; Maimone et al., 2007; Mur-Artal et al., 2015).
	Stereo parallel	Stereo parallel tracking introduces an independent tracking path for the right camera (Zhong & Wildes., 2013). Tracked features are bridged via the left-right matching in the common stereo linear setup.
	Circular	Circular matching traces a feature back to its originating image. For example, a circular matching pattern that tracks the left image's features from Frame $j - 1$ to Frame j, crossing to the right image, back to Frame $j - 1$, and finally back to the left image, was shown to be an effective approach to remove false matching results (Geiger et al., 2011; Ci & Huang, 2016; Min, Xu, Li, Zhang & Zhao, 2016).
	Cross-eye	Cross matching not only avoids occlusion but also offers a better precision in triangulation. Based on such a setup, a multi-camera multi-frame feature integration technique has been recently proposed (Chien, Geng, Chen & Klette, 2016).

recovery is also based on an `OpenCV` implementation.

GPS and IMU data, provided by the dataset, is used as ground truth. The motion-estimation error analysis of the tested tracking topologies is plotted in Fig. 4.9. Referring to the linear topology as the baseline, we found that three configurations show improved performance, up to 30.2%. The best estimation is achieved by the stereo-parallel topology which maintains features from left and right images in parallel, and has them integrated by a left-right matching.

The circular tracking pattern achieves a performance comparable to the cross-eye topology. Both of the tracking patterns reduced motion drift by 15% over a monocular tracking method. Such finding corresponds to previous work (Chien, Geng et al., 2016).

On the other hand, the stereo-linear topology shows a slightly worse performance. A possible cause is due to incorrectly associated left-right features, as in this work we impose only generic epipolar constraints without further limiting the range of left-right disparity.

Another configuration that shows even worse results is found to be the look-ahead topology. We detected numerous incorrectly associated features due to inaccurate optical flow computation over large image displacements, especially in the look-ahead step. As in our current implementation the tracking operator considers only two frames in each step, the third frame does not have any impact. Such a configuration can be further improved by using trifocal tensors for 3-view geometry (Kitt et al., 2010; Hartley & Zisserman, 2004).

4.7 Closure

This chapter presented a novel VO framework. Based on a unifying framework, the proposed implementation achieves accurate egomotion estimation targeting multiple alignment models which previously have been deployed independently in two families of

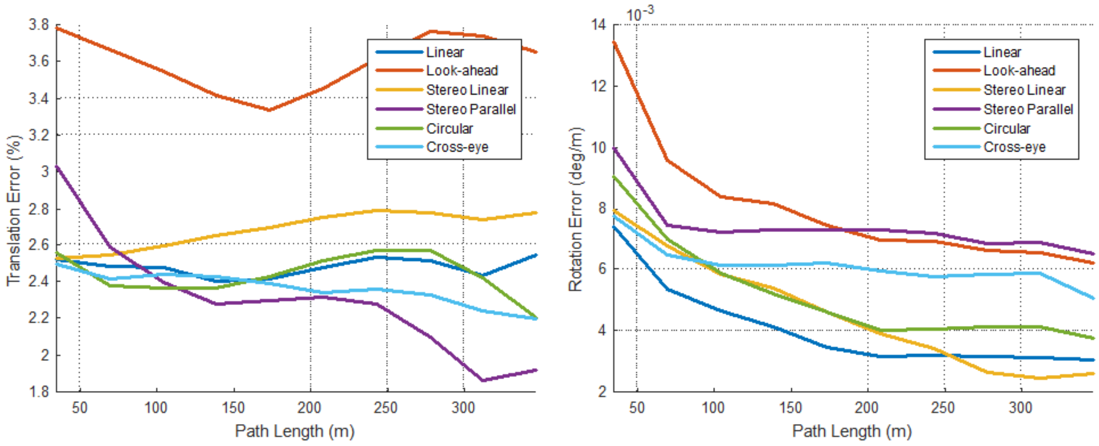


Figure 4.9: Evaluated motion error presented in two components: transitional (*left*) and rotational (*right*). For error analysis, see Appendix B for details.

existing VO methods. Our VO pipeline uses recursive Bayesian filtering to continuously integrate state measurements over time, based on the propagation of error covariances through the underlying VO pipeline. Experimental results show the improvement based on our contributions.

Based on the enhanced egomotion estimation technique, we present a multi-sequence 3D mapper for street-side reconstruction in the next chapter.

Chapter 5

Multiple-run Reconstruction

The integration of reconstructions of the same scene, based on different sequences, can fill-in missing data and improve previous reconstructions. The integration requires a rigid transform between referenced frames of each of the two considered sequences. This chapter presents a solution based on an extension of the unifying VO framework proposed in the previous chapter.

5.1 Overview

Solved egomotion for a given sequence allows us to register 3D data, collected in each frame, in a consistent (world) coordinate system which is essential for global scene reconstruction. Such a reconstruction can be further expanded by merging with another reconstructed scene, assuming that there are overlapping data in both sequences.

This expansion requires that two reconstructions are presented in one consistent coordinate system. A *target sequence*'s frame is considered in the reference coordinate system of the *source sequence*. The problem is equivalent to finding a rigid transform that maps the reconstruction from the source sequence to the target one.

Finding such a transform, however, is not straightforward. One of the simplest solutions is perhaps to treat the problem as registering two point clouds which can be efficiently solved by using one of the ICP algorithms (Besl & McKay, 1992). The algorithms usually work out a good solution when the point clouds are roughly aligned. However, this is often not the case as each video sequence might begin at an arbitrary position in the scene. This issue can be addressed by finding a rough alignment first using 3D features such as the *fast point feature histograms* (FPFH) (Rusu, Blodow & Beetz, 2009).

3D features, extracted from the source and the target point clouds, are extracted and matched in feature space to establish a set of 3D sequence-to-sequence correspondences, based on which a Euclidean transform is efficiently solved by deriving a closed-form solution. The extraction of 3D features, however, requires that the reconstructed point clouds pose similar point densities, and is therefore not applicable to a case where heterogeneous 3D structures are present in source and target (e.g. if the source reconstruction is from a stereo camera while the target from LiDAR scans).

Another issue, limiting the applicability of the 3D alignment approach, is the distortion of point clouds due to accumulated egomotion estimation errors. This factor can quickly become significant if the range of scans extends to up to a few hundreds of metres (Zeng & Klette, 2013).

This chapter presents an approach that establishes inter-sequence image correspondences in a scalable manner. It also discusses how these correspondences are used to register reconstructions in a consistent coordinate system. Section 5.2 introduces two sensor configurations as considered in this work. In Section 5.3 we study the establishment of sequence-to-sequence correspondences in a scalable top-down manner. Section presents a robust nonlinear algorithm to align reconstructions based on established correspondences. Finally, Section 5.5 briefly outlines post processing that filters and merges points in the aligned point clouds.

5.2 Sensor configurations

5.2.1 Stereo sequence

A stereo sequence provides synchronised and rectified images from two calibrated cameras. Each stereo image is converted into a disparity map by a stereo matcher and transformed into 3D points as described in Section 3.2.2. The recovered 3D structure is then used to solve the camera's egomotion detection problem applying either a projective or rigid alignment model as formulated in Section 4.3.2. Stereo vision-based VO is one of the most frequently used SLAM approaches; details of the pipeline are

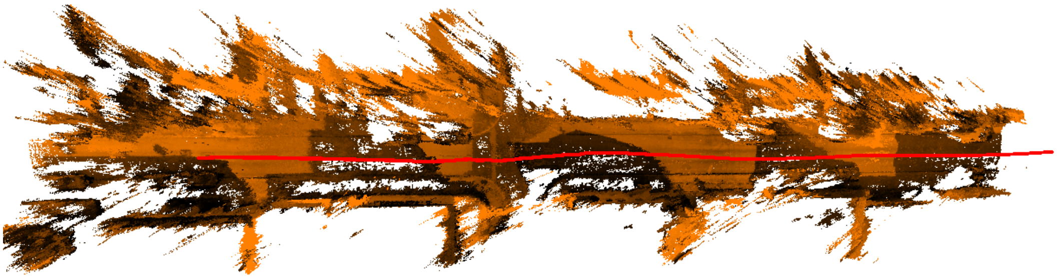


Figure 5.1: Reconstruction of a 120 metres long street-side scene using stereo VO. The imagery data are taken from sequence `Odometry_00` of the KITTI benchmark suite (Geiger et al., 2013).

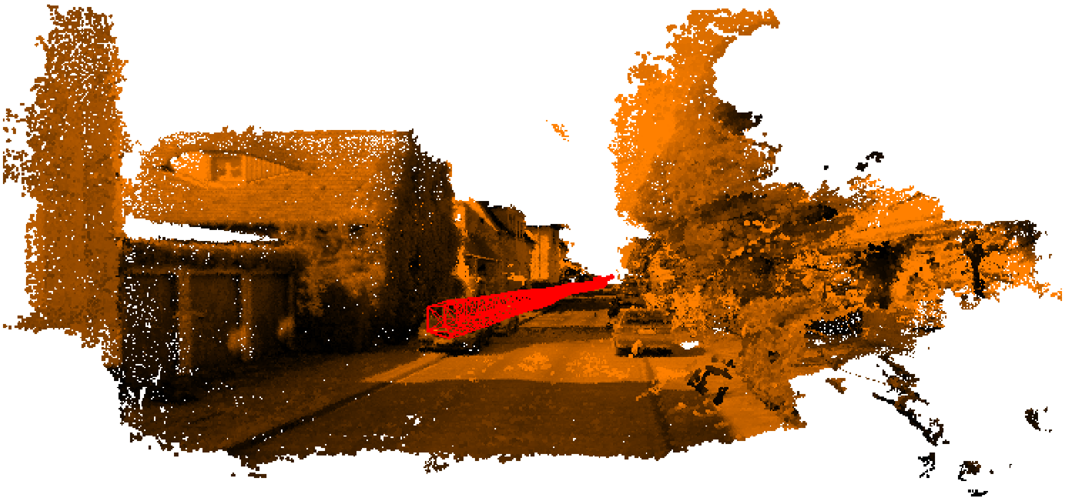


Figure 5.2: Street-view of a the reconstructed scene.

introduced in Chapter 4. Figures 5.1 and 5.2 render 3D reconstruction results of a street-side from a stereo sequence.

5.2.2 Monocular-LiDAR sequence

A few LiDAR-enabled visual odometry systems were proposed in recent years. These systems take the advantage of accurate laser scanning and outperform conventional vision-based odometry methods. By early 2016, the ranking of methods (evaluating odometry accuracy) on the KITTI benchmark website (Geiger et al., 2013) is dominated by LiDAR-based techniques (e.g. (J. Zhang & Singh, 2014, 2015)).

A straightforward implementation of a LiDAR-engaged visual odometry system is to use laser-rendered depth maps for replacing those computed from stereo images. Laser points are projected on the image plane and triangulated to produce an up-sampled dense depth map (Chien, Klette, Schneider & Franke, 2016). See Fig. 5.3 for an example. To enhance the resolution of the rendered depth map, multiple scans are accumulated and aligned using estimated egomotion (J. Zhang & Singh, 2015).

An alternative strategy is to use projections of laser points to establish initial features, track these features to build 3D-to-2D correspondences, and solve egomotion based on following a projective alignment model.

Let $\mathcal{L}_j(\chi)$ be laser-measured 3D coordinates of features χ in the j-th LiDAR frame. The Euclidean measurement function, previously introduced in Section 4.4, is now redefined as follows:

$$g_j(\chi; \mathbf{\Gamma}, \tau) = \mathbf{\Gamma}\mathcal{L}_j(\chi) + \tau \tag{5.1}$$

where g is now parametrised over extrinsic parameters $\mathbf{\Gamma} \in \mathbb{SO}(3)$, and $\tau \in \mathbb{R}^3$.

The image coordinates of χ in Frames $j-1$ and j are then, respectively, decided by

$$\rho_{j-1}(\chi) = \pi\left[g_j\left(\chi; \mathbf{\Gamma}, \tau\right)\right] \tag{5.2}$$

and

$$\rho_j(\chi) = \nu_k(\rho_{j-1}) \tag{5.3}$$

where an image-feature tracker $\nu_k : \mathbb{R}^2 \to \mathbb{R}^2$ is introduced. Parameters Γ and τ are constant if the LiDAR has been extrinsically calibrated and is assumed to be static.

Following these formulations, egomotion is solved based on projective and epipolar alignment models as introduced in Section 4.3. LiDAR-aided PnP ego-motion estimation, described previously, however, needs to be enhanced when applied to real-world sequences where non-static features and noisy tracking results are present.

In this work, we propose a robust monocular-LiDAR implementation by performing image-feature detection and their extraction independently from the LiDAR-engaged framework. The tracked features are maintained by an independent monocular visual odometry module which provides epipolar constraints as well as projective alignment constraints that are derived from those previously triangulated and integrated features

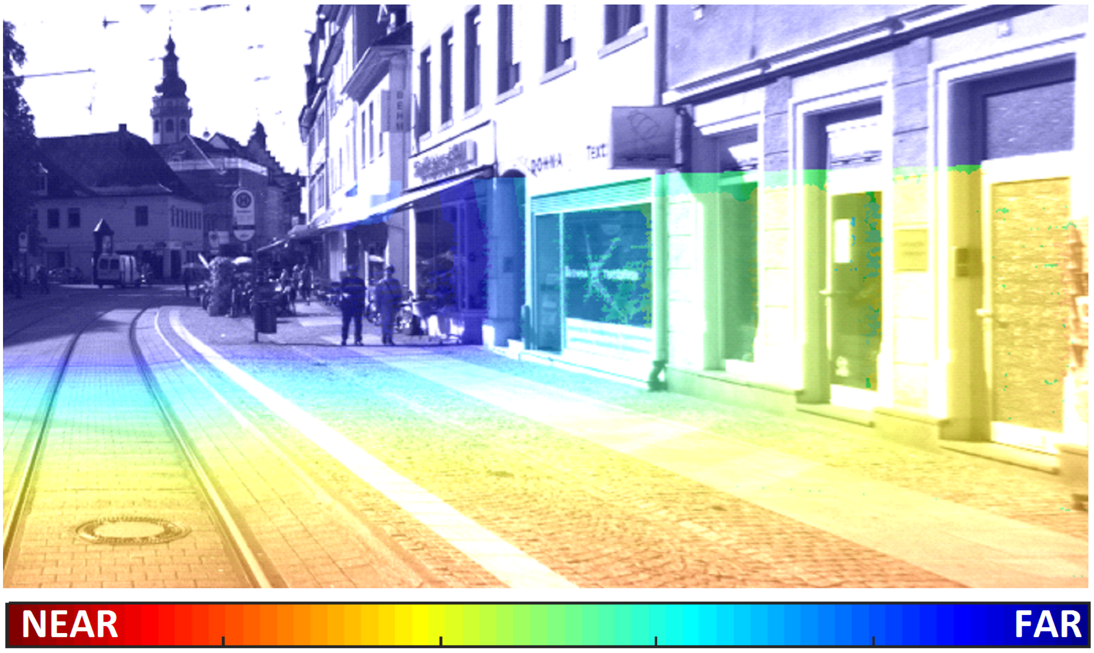


Figure 5.3: Example of a dense depth map from up-sampled LiDAR data. The figure is rendered using data from the frame shown in Fig. 3.4.

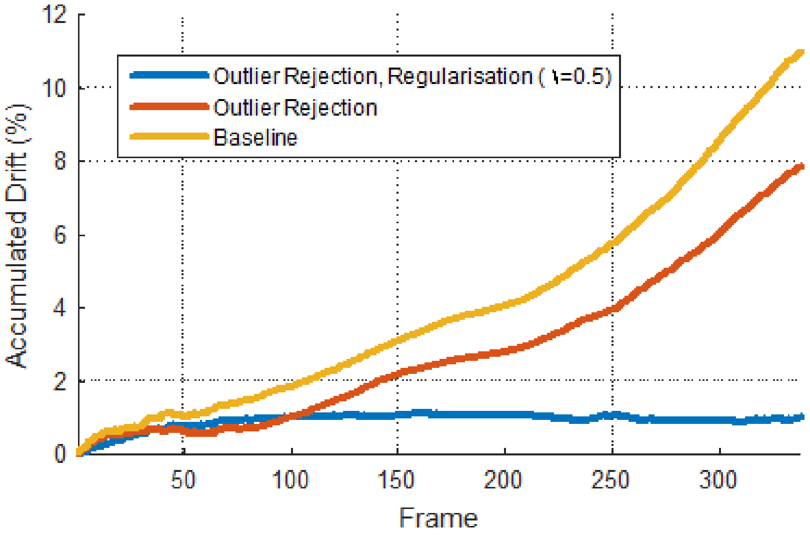


Figure 5.4: Monocular-LiDAR VO framework results for over a 350 metres sequence. The epipolar constrained LiDAR-PnP model, enhanced by an augmented monocular alignment model (as depicted by the blue line), shows a significant reduction in motion drift over implementations either without improvement and monocular vision (in red), or without multi-objective outlier rejection schemes (in orange). See Section 4.2 for the multi-objective model. The evaluation is based on sequence `2011_09_26_drive_0091_sync` of the KITTI Vision Benchmark Suite (Geiger et al., 2013).

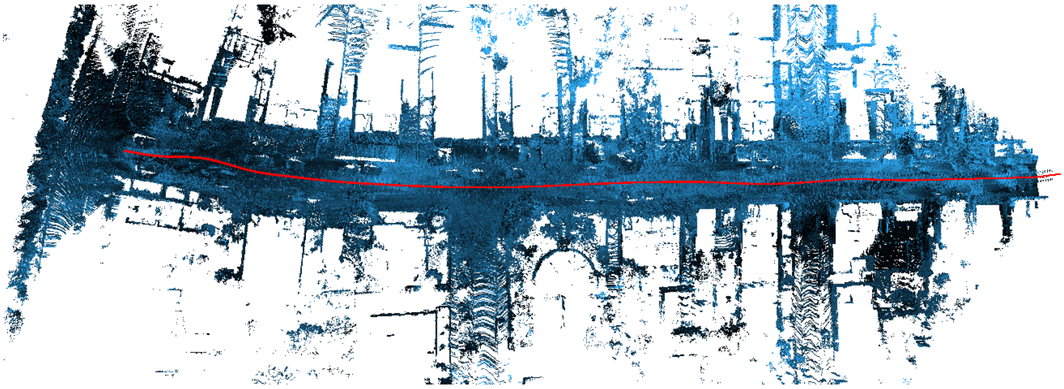


Figure 5.5: Reconstruction of a street side from 250 frames using monocular-LiDAR VO. The range of the first one-third sequence is overlapped with the other sequence shown in Figure 5.1.

(see Section 4.5.2). In this case, the summation of energy models, as described by Eq. (4.22), has to be rewritten as follows:

$$\Phi(\xi) = \Phi_{\text{RPE}}(\xi) + \Phi_{\text{EPI}}(\xi) + \Phi^{\mathcal{L}}_{\text{RPE}}(\xi) + \Phi^{\mathcal{L}}_{\text{EPI}}(\xi)\,. \tag{5.4}$$

where the LiDAR-related models $\Phi^{\mathcal{L}}_{\text{RPE}}$ and $\Phi^{\mathcal{L}}_{\text{EPI}}$ are appended after the monocular ones.

Figure 5.4 shows that the drift of monocular-LiDAR VO can be effectively reduced by considering all four terms in Eq. (5.4), compared to using only one single model. Figure 5.5 visualises the reconstruction of the same scene previously shown in Fig. 5.1 from another longer run.

5.3 Establishing inter-sequence correspondences

Reconstructions from multiple runs need to be aligned in a consistent coordinate system to allow us that depth data are properly integrated. In this section we propose a scalable solution that first selects, from each sequence, a set of *keyframes*, where subsequent keyframes contain a minor amount of overlapping data, such that the computation burden of cross matching at latter stages is reduced.

Based on these extracted keyframes we deploy then an efficient bag-of-visual-word technique that transforms each frame into a vector representation, known as the *vocabulary*. A brute-force search on the vocabulary is conducted to identify potential cross-matches among keyframes, between source and target sequences.

At the final stage, the landmarks of any two matched keyframes are then associated in the descriptor space, and their 3D positions as well as their 2D projections are used to build a projective alignment-based egomotion model following the framework presented in Chapter 4.

Based on this model, a rigid transform is estimated in RANSAC manner which contains essential information to align the source sequence to the target's frame.

5.3.1 Keyframe extraction

Key-framing is an efficient technique to build a compact skeleton of a long image sequence by selecting a subset of frames that essentially contain sufficient information of the sequence. There are many possible criteria for keyframe selection. The selection can be based on, for example, a fixed time interval, the distance the camera has travelled since the last keyframe, or the number of lost features.

In this work we select keyframes based on *covisibility* which is defined by the number of shared landmarks between two frames. The visibility of Landmark i in Frame j is represented formally as follows:

$$v_{ij} = \begin{cases} 1, & \text{if the } i\text{-th landmark is observed in the } j\text{-th frame} \\ 0, & \text{otherwise}\,. \end{cases} \tag{5.5}$$

Based on Eq. (5.5), the covisibility is defined among Frames j and j' as follows:

$$\text{covis}(j, j') = \frac{\sum_i v_{ij} \cdot v_{ij'}}{\sum_i v_{ij}} \tag{5.6}$$

See Fig. 5.6 for an example of a visualised covisibility matrix.

If the covisibility among the current frame and the closest keyframe reaches a lower bound, it implies that many features have been lost, and it is a good time to select the current frame as a new keyframe. In the case of VO, the images are considered to be collected in temporal order, therefore we only need to consider the covisibility between the current frame and the previous keyframe.

Covisibility-based keyframe selection is more robust than applying motion-based criteria which might fail when the sensor presents a large variety in movement patterns.

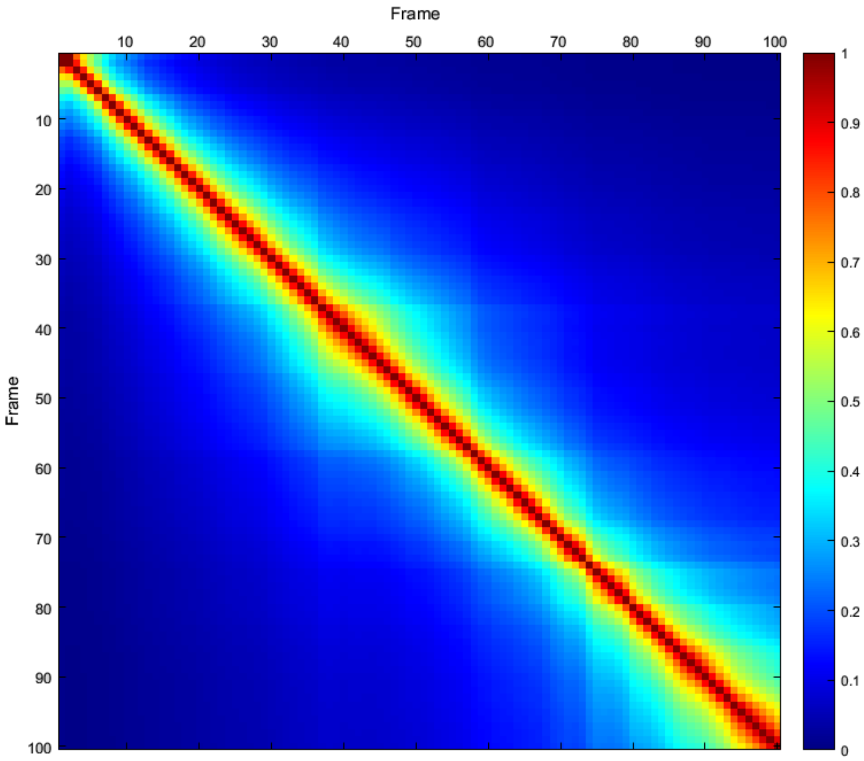


Figure 5.6: Covisibility matrix of the first 100 frames of the typical street sequence shown in Fig. 5.1. Note that the locality pattern constantly changes due to the motion of the camera.

Figure 5.6 shows the covisibility of a number of continuous frames; results illustrate (when compared with the recorded sequences) that non-uniform locality patterns highly dependent on the dominating motion component of the sensor's egomotion as well as on scene structure.

The snapshots of 24 keyframes, extracted from Fig. 5.6 are shown in Fig. 5.7, where the minimum covisibility between any two consecutive keyframes is set to be 0.3.

5.3.2 Bag-of-words model

The *bag-of-words* (BoW) model was originally developed in natural language processing and information retrieval. It has recently been extended to SLAM for recognition of previously visited places (Galvez-López & Tardos, 2012; Lowry et al., 2016). The BoW

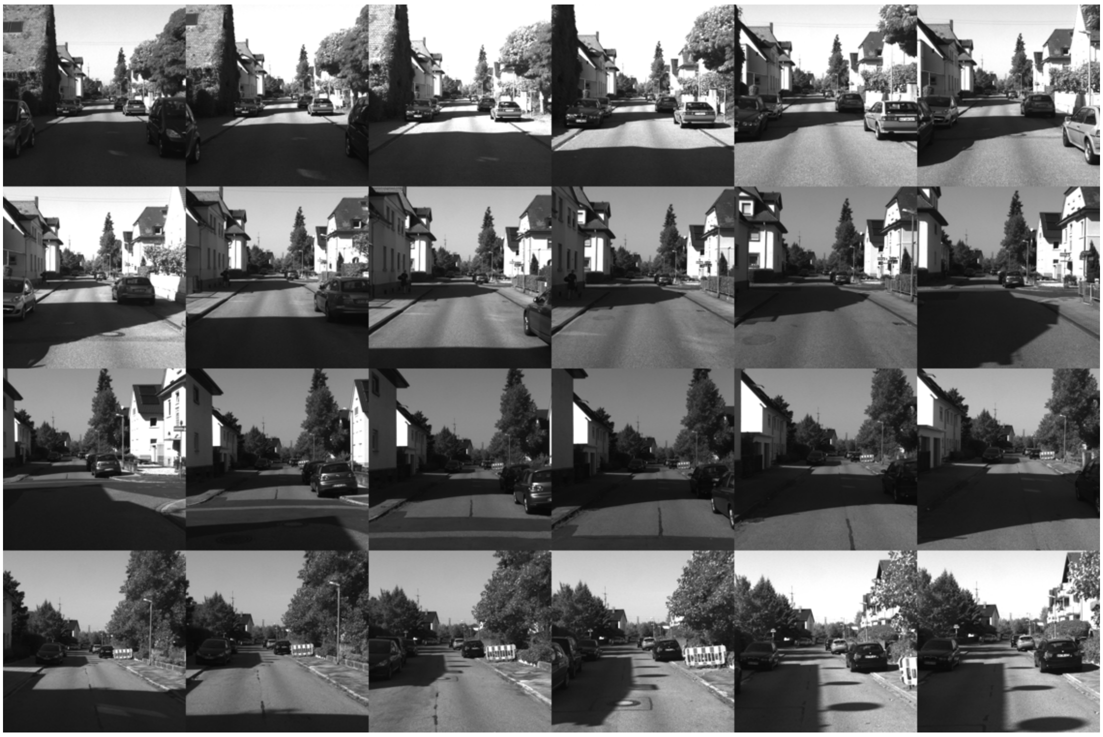


Figure 5.7: Extracted keyframes with a minimum covisibility of 0.3.

model identifies from documents a set of keywords (the *vocabulary*), based on which each document is transformed into a vector representation in the frequency domain. Two documents, sharing similar components in the domain, are classified into the same category in the context of document classification.

The extension of the BoW model to computer vision considers an image as being a document, and features in the image are *visual words*. To generate a visual vocabulary, descriptors of image features, extracted from a set of training images, are clustered in descriptor (or feature-vector) space by means of a vector quantisation technique (e.g. use of *k-means* clustering). The mean of each learned cluster is then chosen for generating the vocabulary. The generation is done offline, using images that contain a rich variety of appearances.

We follow an efficient BoW technique when detecting keyframes that were possibly collected at the same place (Galvez-López & Tardos, 2012). A set of keywords is trained offline using *oriented FAST and rotated BRIEF* (ORB) features (see Appendix B.1.3 for details).

The generated vocabulary is organised as an L-level K-ary tree, with the visual keywords stored at leaf level, where $L = 10$ and $K = 5$ in our case. Learned keywords are further weighted by *term frequency-inverse document frequency* (tf-idf) index, in a way such that rare words are given higher weights as they are more discriminative.

To establish frame-to-frame correspondences between two sequences, we first transform each keyframe, extracted from the source sequence, into a frequency-based vector representation. This is done by finding ORB features from the frame and mapping extracted binary descriptors onto the learned W vocabulary.

The resulting vector has W entries, with the i-th entry storing the weighted occurrences of Keyword i in the keyframe. An inverse index database is built from the transformed source keyframes. The database maintains a list of keyframes that share the same keyword.

In the inquiry phase, each keyframe from the target sequence is transformed into its vector representation following the same process applied to the source keyframes.

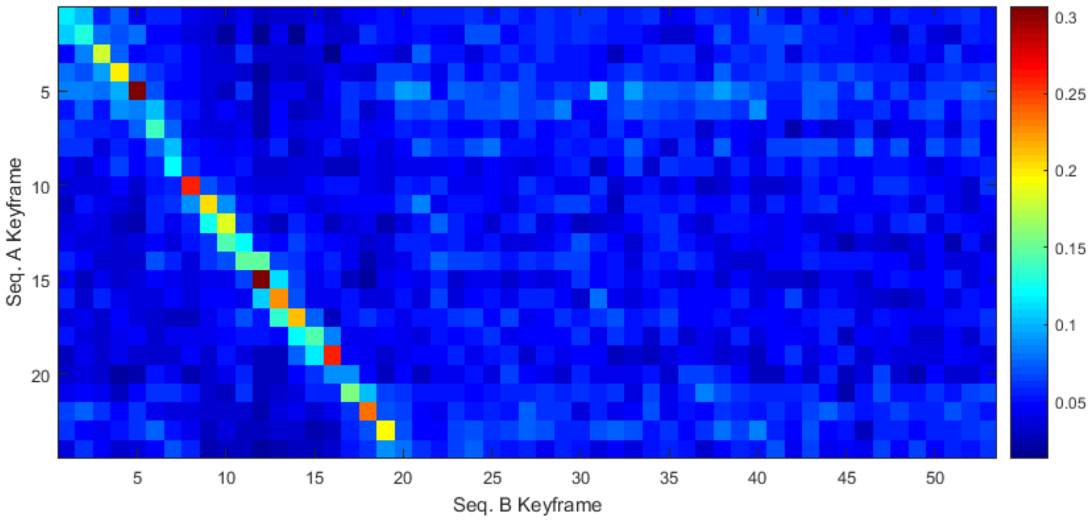


Figure 5.8: Cross-sequence similarity matrix, computed using $14,000$ binary visual words based on ORB features extracted from a source sequence's keyframes.

The database is then searched to conclude similarity between a target keyframe and any of the source keyframes. The search is computationally scalable as the complexity is logarithmically bound by the size of the database, thanks to the tree-structured inverse indices and the sparse vector representations.

Figure 5.8 shows a similarity matrix that takes 24 keyframes of a stereo sequences (see Fig. 5.1) as the source, and 53 keyframes of a monocular-LiDAR sequence (see Fig. 5.5) as the target. The computation was done in real-time on a mainstream quad-core laptop. Results show strong connections between the source sequence and the first 19 target keyframes.

Two keyframes are considered to be *matched* if the similarity is higher than a threshold, say 0.2 in this work.

5.3.3 False-positive rejection

A cross-sequence correspondence hypothesis, suggested by the BoW model, might be a false positive as the spatial relationships among features are not taken into account throughout the process so far. This therefore requires further checks on geometric constraints.

In this work we use a fundamental matrix-based outlier rejection model for a fast rejection of any false-positive match. Note that the constraint, introduced by a fundamental matrix, is not a strict criterion at this stage, where the sensors' intrinsic parameters are available to allow us a rigorous check. However, we leave such a check to the next stage where outliers are examined again by a strict alignment model.

5.4 Aligning sequences

Based on discovered cross-sequence, frame-to-frame correspondences, a rigid transform is initially estimated using a RANSAC process by means of a multi-projective alignment model. The process is followed by keyframe-based bundle adjustment for refining the alignment of the sequences.

5.4.1 Finding an initial pose

For each frame-to-frame correspondence, a set of finer correspondences is established at the landmark-to-image level. Descriptors of the landmarks, previously observed in the source frame at the VO stage, are matched with those observed in the target frame. A set of forward and backward projective constraints is then built following the egomotion-estimation strategy described in Section 4.4.

Taking into account the poses of keyframes, solved in the VO process, all the discovered constraints are aggregated and transformed into a coherent coordinate system centred at the reference frame of the target sequence. This way, a solution, minimising the resulting energy model, serves as the initial pose of the source sequence's reference frame with respect to the target one's. Here, we adopt the same RANSAC technique as shown in the previous chapter for solving for the pose.

5.4.2 Refinement using bundle adjustment

Bundle adjustment (BA) is considered to be the "golden standard" optimisation technique already for multi-view reconstruction over decades of research. The technique simultaneously tunes camera parameters and scene structure to fit a nonlinear function, in a way that the discrepancy between the observations of landmarks and their reprojections are minimised in a least-squares manner.

Given M landmarks observed in N frames, the objective function specific to the BA problem in an abstract form is defined as follows:

$$\Phi_{\text{BA}}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{M} \sum_{j=1}^{N} v_{ij} \|\mathbf{y}_{ij} - f(\mathbf{a}_{\mathrm{j}}, \mathbf{b}_{\mathrm{i}})\|_{\mathbf{\Sigma}_j}^2 \tag{5.7}$$

where $\mathbf{y}_{ij}$ are the image coordinates of the i-th landmark observed in the j-th frame, $\mathbf{\Sigma}_j$ is the 2×2 covariance matrix modelling the uncertainty of the observation, $\mathbf{a}_j \in \mathbb{V}_a$ and $\mathbf{b}_i \in \mathbb{V}_b$ are, respectively, parametric forms of a camera pose and a 3D point in the chosen parameter spaces $\mathbb{V}_a$ and $\mathbb{V}_b$, v_{ij} is the binary function denoting landmark's visibility as defined in Section 5.3.1, and $f : \mathbb{V}_a \times \mathbb{V}_b \to \mathbb{R}^2$ is the abstracted imaging function.

As discussed in Section 3.4.4, an objective function in the least-square form, posed by Eq. (5.7), can be minimised using the Levenberg-Marquardt (LM) algorithm (Levenberg, 1944).

Let $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ where $n = \dim(\mathbb{V}_a) \cdot N + \dim(\mathbb{V}_b) \cdot M$ be the state vector that vectorises $\mathbf{a}$ and $\mathbf{b}$. Then, for m nonzero components, Eq. (5.7) can be represented in a more compact form by a vector function $\varphi : \mathbb{R}^n \to \mathbb{R}^m$ where each entry $\varphi_k(\mathbf{x})$ denotes the residual of the observation k and its projection. This way, minimising Φ_{BA} is equivalent to minimising the L_2-norm of φ (i.e. $\|\varphi\|^2$) when identity covariance matrices are assumed.

The LM algorithm uses the Jacobian-approximated second-order derivative of φ to update x iteratively. For each iteration, the update is given by

$$\Delta \mathbf{x} = (\mathbf{J}^{\top}\mathbf{J} + \lambda \mathbf{I})^{-\mathbf{1}} \mathbf{J}^{\top} \varphi(\mathbf{x}) \tag{5.8}$$

where $\mathbf{J}_{ij} = \dfrac{\partial \varphi_i}{\partial x_j}$ is the Jacobian matrix, and $\lambda \in \mathbb{R}_+$ the damping factor (Levenberg, 1944).

If $\mathbf{x}' = \mathbf{x} + \boldsymbol{\Delta}\mathbf{x}$ reduces $\|\varphi\|^2$, the update is accepted and λ is decreased, leading to a Gauss-Newton-like behavior. Otherwise, λ is increased to resemble a gradient descent approach, and a new $\boldsymbol{\Delta}\mathbf{x}$ is solved and tried repeatedly, until it attains a better solution.

The linear system $\mathbf{A} = \mathbf{J}^{\top}\mathbf{J} + \lambda \mathbf{I}$ poses a sparsity pattern due to mutually irrelevant structure parameters. By splitting up the camera and structure components from the linear system

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^\intercal & \mathbf{V} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x_a} \\ \Delta\mathbf{x_b} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_a^\intercal \\ \mathbf{J}_b^\intercal \end{bmatrix} \varepsilon(\mathbf{x}) \tag{5.9}$$

the matrix $\mathbf{V}$ is found; it contains nonzero entries only in its diagonal M sub-matrices of size $\dim(\mathbb{V}_b) \times \dim(\mathbb{V}_b)$. Multiplying Eq. (5.9) by the matrix

$$\begin{bmatrix} \mathbf{I} & -\mathbf{W}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{5.10}$$

yields

$$\begin{bmatrix} \mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^\intercal & \mathbf{0} \\ \mathbf{W}^\intercal & \mathbf{V} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x_a} \\ \Delta\mathbf{x_b} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_a^\intercal - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^\intercal\mathbf{J}_b^\intercal \\ \mathbf{J}_b^\intercal \end{bmatrix} \varepsilon(\mathbf{x}) \tag{5.11}$$

where $\Delta\mathbf{x_a}$ is first solved and then $\Delta\mathbf{x_b}$.

By exploiting the blocky sparse structure of $\mathbf{V}$, its inverse can be quickly computed in linear time $\mathcal{O}(M)$. Solving $\Delta\mathbf{x}$ in this two-stage fashion greatly alleviates the computational complexity of BA, since $\mathbf{A}$ is typically a giant matrix that involves a huge number of scene points but only a few camera parameters (Lourakis & Argyros, 2009; Konolige, 2010). A runtime breakdown of solving BA problems for various sizes is given in Fig. 5.9.

The breakdown shows that the computation time is dominated by the number of landmarks involved in the adjustment. In the case of aligning two sequences, it often covers thousands to millions of covisible landmarks. Under such circumstances, the optimisation is time consuming.

In this research, an algebraic technique is deployed to address such issues. Recall the linear triangulation approach described in Section 3.2.1. The central-projection model relates camera pose and scene structure linearly in the projective space. Given a 3-by-4 projection matrix $\mathbf{P}$, a scene point $\tilde{\mathbf{x}}$, and its image $\tilde{\mathbf{y}}$, both in homogeneous coordinates, follow the relationship

$$\tilde{\mathbf{y}} \sim \mathbf{P}\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 \end{bmatrix}^{\mathsf{T}} \tilde{\mathbf{x}} \tag{5.12}$$

where ~ denotes the equality up to a scale. Alternatively, in homogeneous form, this relationship is expressed as follows:

$$\begin{bmatrix} y_3\mathbf{p}_1^{\mathsf{T}}\tilde{\mathbf{x}} - y_1\mathbf{p}_3^{\mathsf{T}}\tilde{\mathbf{x}} \\ y_3\mathbf{p}_2^{\mathsf{T}}\tilde{\mathbf{x}} - y_2\mathbf{p}_3^{\mathsf{T}}\tilde{\mathbf{x}} \end{bmatrix} = 0\,. \tag{5.13}$$

This equation, posing a linear duality between the structure parameters and the parameters of the camera, can be inherently extended to a multi-view case where the 3D coordinates of a landmark can be immediately determined given its observations in more than two frames.

Using the duality, we embed the estimation of structure in the adjustment of camera pose during the BA process. This not only eliminates the time-consuming construction of $\mathbf{J}_\mathrm{b}$, but also removes the impact of a structure-measurement error.

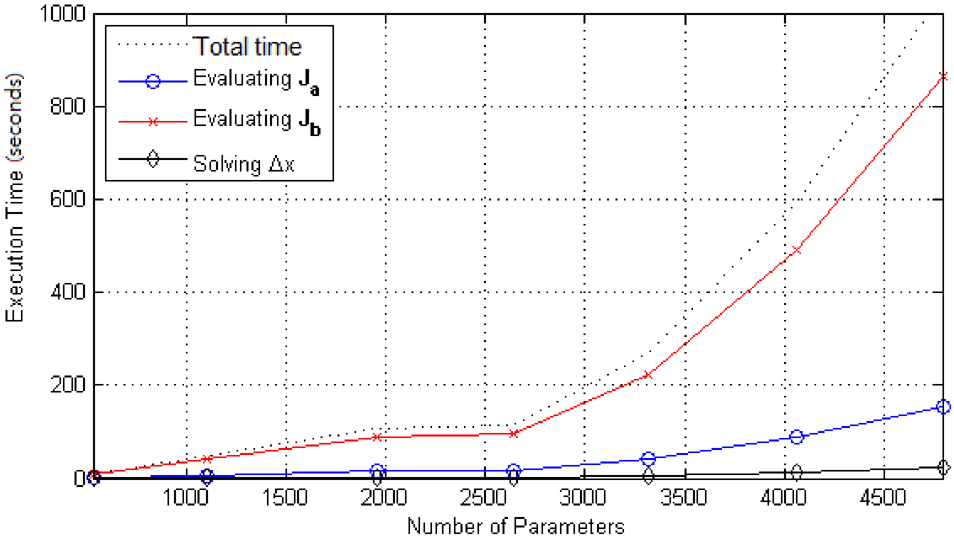


Figure 5.9: Comparison of computation time of a bundle adjustment process.

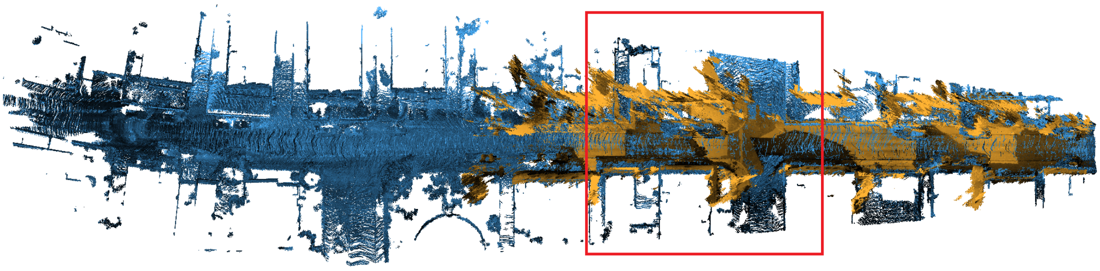


Figure 5.10: Visualisation of two sequences posed in their optimised positions after keyframe-based BA. The sequences are identified by different colours, as previously rendered in Figs. 5.1 and 5.5. A closer look at the marked area is provided in Fig. 5.11.

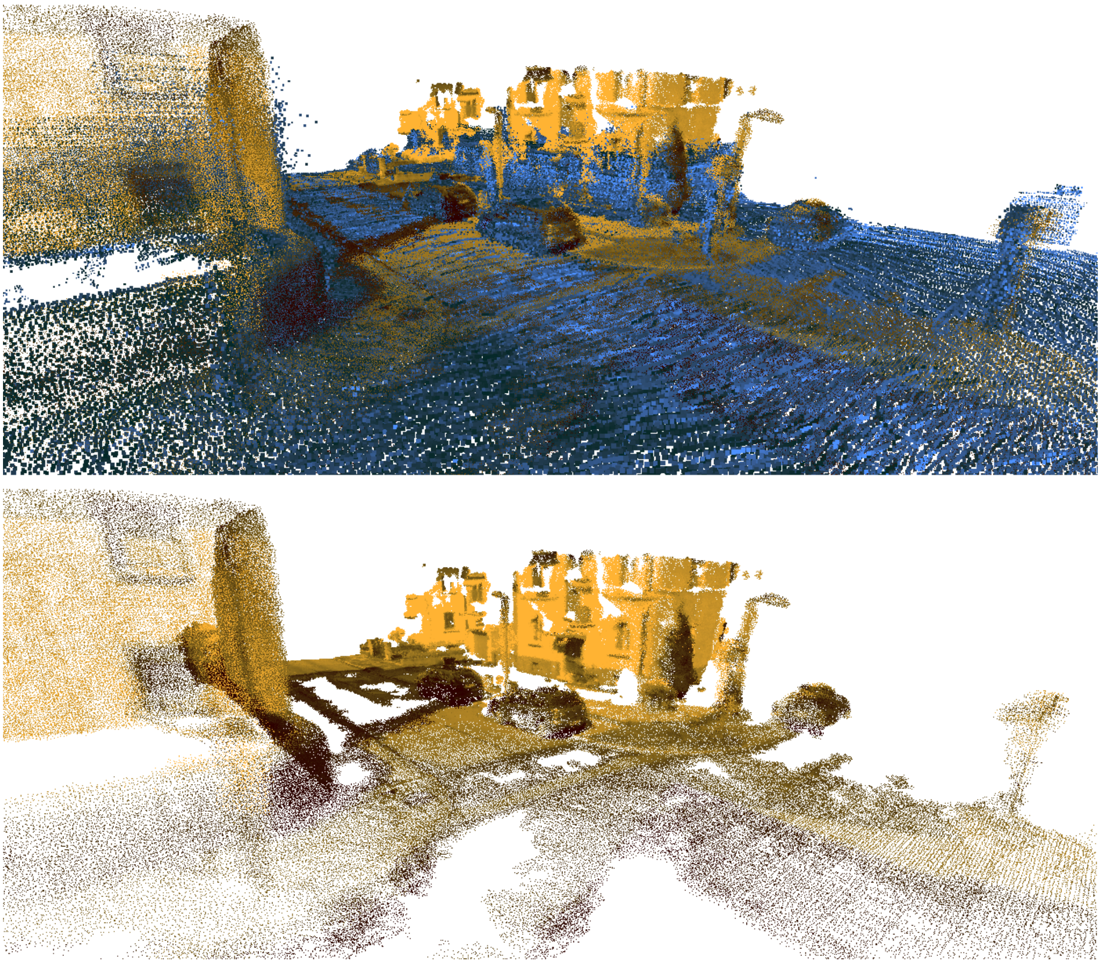


Figure 5.11: *Top:* A closer look on the aligned reconstruction at a cross-section. *Bottom:* The reconstruction from a stereo sequence at the same location. Much information of the road surface is missing from the disparity map due to low confidence. Laser scans from a different run filled in the missing information as can be seen in the figure.

The landmarks, mutually observed in the keyframes of both sequences, as well as the pose of these keyframes are selected for a nonlinear adjustment, with the landmark structure implicitly modelled. The BA process is bootstrapped by the pose estimated using the RANSAC approach as introduced in the previous section.

Aligned frames are visualised in Figure 5.10, and a closer look on the aligned sequences is shown in Figure 5.11. It can be found that, a great portion of missing data from the first run is filled by the second run.

5.5 Depth integration and surface reconstruction

Once two sequences are aligned, we construct a 3D point cloud from back-projected disparity maps from the stereo sequence and LiDAR scans of a monocular LiDAR sequence. Considering the huge number of vertices in the noisy clouds, we perform cleaning and subsampling pre-processing prior to depth integration and surface reconstruction.

Isolated floating points, that have only a few neighbours, are identified as being noise and are removed. Vertices closer than a predefined threshold, say 0.15 metres, to each other are merged for reducing the size of the point cloud. The vertex normal vectors are then estimated from the filtered point cloud. For normal computation we select near neighbours of each vertex, and a least-squares tangent plane is estimated (Mitra, Nguyen & Guibas, 2004).

A variation of a *Poisson surface reconstruction* is adopted to triangulate[1] the processed vertices. Given a discrete set of 3D points $P = \{\mathbf{p}_1, \mathbf{p}_2, ..\mathbf{p}_N\}$ equipped with normal vectors $V = \{\mathbf{n}_1, \mathbf{n}_2, ..\mathbf{n}_N\}$, surface reconstruction aims at finding a scalar function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ such that $f(\mathbf{p}) = 0, \forall \mathbf{p} \in P$ and $\nabla f(\mathbf{p}) = \mathbf{n}, \forall \mathbf{n} \in V$, where $\mathbf{p}$ is the

[1] The triangulation here refers to building a triangular mesh from given sparse vertices.

point corresponding to $\mathbf{n}$. Modelling such a problem by a sum-of-squares minimiser

$$\varphi(f) = \sum_i \|\mathbf{n_i} - \nabla f(\mathbf{p}_i)\|^2 \tag{5.14}$$

leads to a least-squares solution in its second derivative form

$$\Delta f(\mathbf{p}_i) = \nabla \mathbf{n_i} \tag{5.15}$$

where Δ is the Laplacian operator, and $\nabla \mathbf{n_i}$ is the gradient of a vector field constructed from V at $\mathbf{p}_i$. This partial differential equation is known as *Poisson equation*.

A finite solution to Eq. (5.15) is based on a discretization of Poison's equation which first selects a set of N B-splines $\{g_1, g_2, ..., g_N\} : \mathbb{R}^3 \to \mathbb{R}$ to form the new basis of f:

$$f(p) = \sum_{j=1}^{N} x_j g_j(p)\,. \tag{5.16}$$

Let $\langle \square, \square \rangle$ be the standard inner product. By Eq. (5.15) we have that

$$\langle \Delta f(\mathbf{p}_i), \nabla g_j(\mathbf{p}_i) \rangle = \langle \mathbf{n_i}, \nabla g_j(\mathbf{p}_i) \rangle \tag{5.17}$$

An over-determined linear system of M unknowns is instantiated by taking into account all the $(\mathbf{p}_i, \mathbf{n}_i)$'s and g_j's.

In this work we selected an efficient octree-based hierarchical solver to approach a discrete solution to the linear system (Kazhdan & Hoppe, 2013). Once the weights $x_1, x_2, ...x_j$ are found, the mesh reconstruction is achieved by an iso-surface extraction at $f = 0$. Figure 5.12 visualises an iso-surface consisting of 1.9 millions of triangles having about $965,000$ vertices. The reconstruction is based on a segment of the merged sequences.

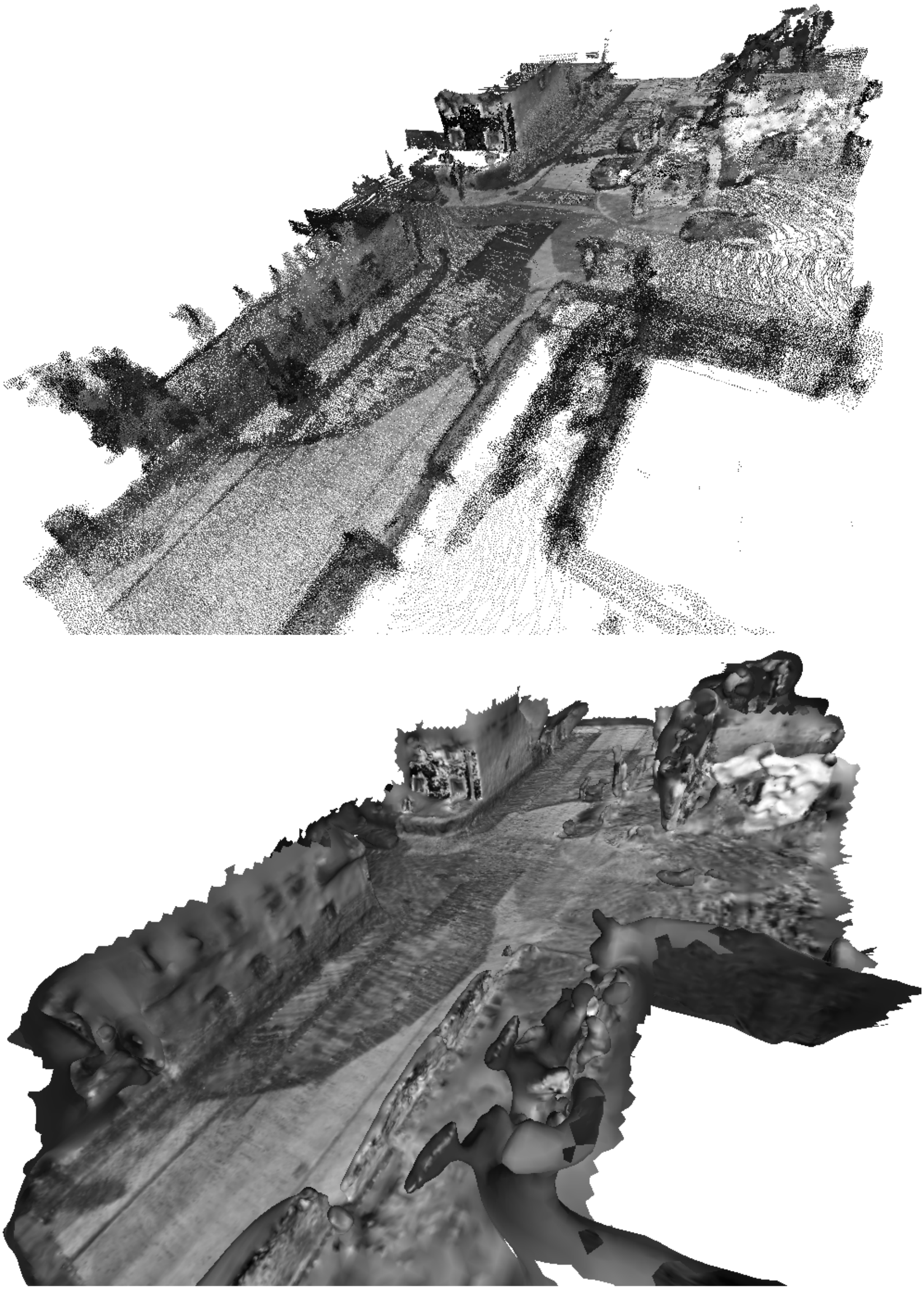


Figure 5.12: *Top:* A filtered point cloud from two merged sequences. *Bottom:* The reconstruction of the point cloud using a Poisson surface reconstruction process.

5.6 Conclusion

In this chapter we presented a novel strategy to combine sequences of multiple runs, generated by different sensor configurations, to produce a 3D reconstruction of a street-side scene. The sequences are initially aligned using landmark-feature correspondences discovered among keyframes. A nonlinear adjustment is then carried out to optimise the poses of matched keyframes. Once aligned, the 3D data of the scene structure, from each sequence, is processed and integrated in a consistent coordinate system, resulting in a comprehensive reconstruction of the street-side scene.

Chapter 6

Future Work

The work presented in this thesis can be further improved and extended toward a number of directions. The unifying VO framework, introduced in Chapter 4, has been tested using binocular stereo vision as well as monocular-LiDAR configurations. The model can be extended in a straightforward way to a trinocular setup which provides some interesting options in terms of feature tracking topology. The third camera also provides an additional source for verification of the solved egomotion (Chien, Geng & Klette, 2014). A calibration tool for trinocular setup is readily deployed, as tested in Appendix A.

We noticed that many buildings in reconstructed scenes are not yet complete (i.e. their 3D surface). Perhaps such missing pieces can be filled by using a drone flying around such buildings. If this provides an additional monocular sequence in this case, the sequence aligning pipeline, introduced in Section 5.4, needs to be reworked to consider the scale of the monocular reconstruction as an unknown factor, meaning that the solution to the alignment is generalised from a 6-dof Euclidean transformation to a 7-dof similar one.

It is interesting to study the registration of sequences collected from opposite directions, which remains challenging. In Fig. 6.1, two frames acquired at the same

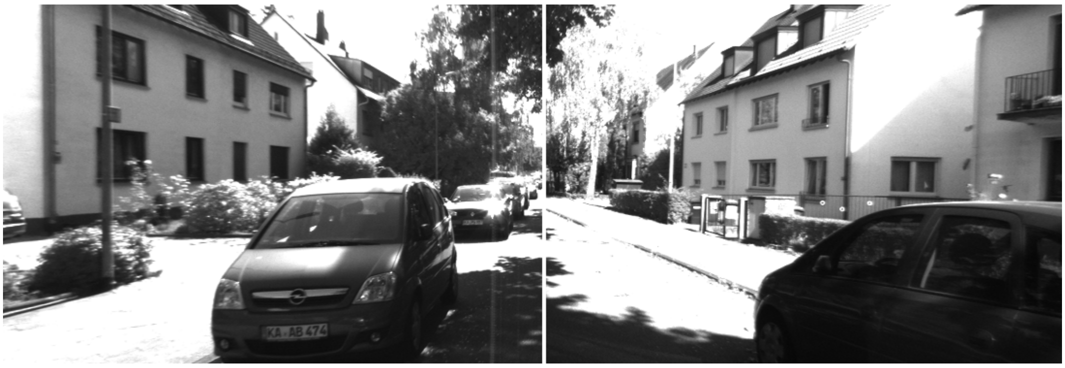


Figure 6.1: Two image sequences collected from opposite directions on which the BoW model failed to establish frame-to-frame correspondences.

place but in very different viewing directions, failed to be associated using the BoW model described in Section 5.3.2. The issue might be addressed using features in 3D space to establish an initial alignment (Zeng & Klette, 2013), given that the estimated camera poses are accurate to avoid distorted point clouds integrated over multiple frames.

Applying the proposed integration technique to more than just two sequences also poses new issues that are worth studying. For many-to-many alignments, the keyframe-based search scheme, adopted in Section 5.3.2, would be less efficient. An incremental computational-efficient approach needs to be developed. Identification and removal of temporal objects (e.g. parked cars), that are not persistent in multiple sequences, is also required to achieve consistent street-side reconstruction.

In many applications it is desired to access high-level information from a reconstruction for better scene understanding (Badino, Franke & Pfeiffer, 2009). In that case, the scene can be modelled at a higher level, based on aligned point clouds and recovered surfaces (N. Savinov & Pollefeys, 2016).

Appendix A

Camera Model and Multiocular Calibration

Calibration of geometric imaging parameters is a fundamental task in the field of computer vision. Obtaining the imaging parameters allows accurate modeling of the incident ray of each pixel, and once backprojected the light paths can be used to estimate 3D structure of the scene.

Calibrating an image sensor is essential to find the inverse mapping from image space to world coordinates, given a set of 3D-to-2D correspondences $(X, Y, Z) \to (x, y)$. Any object that once imaged produces such correspondences can be selected as a calibration target. In this work we consider the use of a planar chessboard as the target, as the making of such target is easier and the manufacturing accuracy can be controlled.

Off-the-shelf camera calibration packages such as `OpenCV` `calib3d` module (“The OpenCV Reference Manual: Camera Calibration and 3D Reconstruction”, 2017) and MATLAB Camera Calibration Toolbox (Bouguet, 2015) are available. These tools, however, are originally designed for monocular and binocular configurations, and their straightforward extension to multiocular case leads to consistency issues. This appendix develops an automatic calibration algorithm that can be applied to monocular, binocular and multiocular cases.

A.1 Nonlinear camera model

We follow the nonlinear camera model implemented by `OpenCV 3.3.0` (“The OpenCV Reference Manual: Camera Calibration and 3D Reconstruction”, 2017). Given $\mathbf{E} = (\mathbf{R}, \mathbf{t})$ the extrinsic parameters consist of rotation matrix $\mathbf{R} \in \mathbb{SO}(3)$ and translation vector $\mathbf{t}$, a 3D point (X, Y, Z) is first projected onto the ideal normalised image plane at $Z = 1$ by

$$\begin{pmatrix} \mathring{x} \\ \mathring{y} \\ 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{A.1}$$

where $\sim$ denotes the projective equality (i.e. equivalent up to a non-zero scale). Then the nonlinear distortion applies to find the distorted pixel $(\breve{x}, \breve{y})$:

$$\begin{pmatrix} \breve{x} \\ \breve{y} \\ 1 \end{pmatrix} \sim \begin{pmatrix} \mathring{x} & 2\mathring{x}\mathring{y} & r^2 + 2\mathring{x}^2 & 0 \\ \mathring{y} & r^2 + 2\mathring{y}^2 & 2\mathring{x}\mathring{y} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 \\ p_1 \\ p_2 \\ 1 \end{pmatrix} \tag{A.2}$$

where $r^2 = \mathring{x}^2 + \mathring{y}^2$ is the radial distance, and parameters $\kappa_{1..3}$ and $p_{1..2}$ respectively control the radial and tangential distortion of the image sensor. These parameters denote the distortion coefficients. Finally the distorted pixel is transformed to the observed pixel coordinates

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & 0 & x_c \\ 0 & f_y & y_c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \breve{x} \\ \breve{y} \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \breve{x} \\ \breve{y} \\ 1 \end{pmatrix} \tag{A.3}$$

where f_x and f_y are the effective focal lengths in longitude and latitude directions respectively and (x_c, y_c) is the principal point where optical axis passes through the image plane.

Note that, the entry k_{12} of the intrinsic matrix is set zero. This entry controls the skewness factor of image formation, and is modeled when $\mathbf{K}$ is, more generally, an upper triangle matrix (Hartley & Zisserman, 2004). Alternatively, the skewness factor is modeled here by tangential parameters p_1 and p_2.

A.2 Monocular and stereo calibration

In the very beginning of the development, the direct linear transform (DLT) technique was adopted to solve the calibration problem before any sophisticated algorithms are proposed (Hartley & Zisserman, 2004). The DLT method treats the imaging process as a linear transform and solves the projection matrix directly. Post-processing is therefore required to extract the intrinsic and extrinsic parts of the parameters from the estimated matrix.

In 1987, R. Tsai proposed perhaps the earliest popular camera calibration algorithm (Tsai, 1987). Credited by his name the method has been well-known as *Tsai's method* by the community. The method first calibrates camera's intrinsic parameters and the x, y components of extrinsic parameters, then the z components is estimated by enforcing orthonormality. Tsai's method is able to calibrate a camera from a single image of a 3D calibration object.

A decade later, Z. Zhang published an approach that soon became the de facto standard based on the image of an abstract geometry entity - the *absolute conic* (Z. Zhang, 2000). Given a homography defined by the calibration plane, the orthogonality and normality of the rotation matrix respectively determine two out of six degrees of freedom of the image of the absolute conic (IAC). Since the IAC is defined by the camera's intrinsic parameters and is invariant to the derived homography, a linear system can be constructed by stacking constrains of many (at least 3) observed homographies. Zhang's method first recovers the intrinsic parameters by solving the IAC, then the extrinsics are determined using the solved intrinsics.

In practice the parameters estimated using any of the described techniques are never used directly. Instead, they serve as an initial guess for the nonlinear optimisation stage carried out right after applying a chosen calibration algorithm.

The monocular calibration is extended to calibrate stereo camera rigs. Provided by the `OpenCV` library the subroutine `stereoCalibrate` implements a state-of-the-art stereo calibration used widely by the community. The algorithm first independently calibrates each camera's intrinsic parameters and the pose of the calibration target in each view. Since the target's poses estimated from the first and the second cameras can be inconsistent, the algorithm chooses the first camera-derived pose as the reference, and the target pose derived from the second camera is used only for the estimation of extrinsic parameters. Again, one has to work out the inconsistency of the estimated extrinsic parameters. A workaround to alleviate the impact of inconsistency as implemented by the subroutine is to decide the extrinsic parameters by averaging.[1]

Obviously, several drawbacks exist in the stated algorithm. First, the target has to be observed by all cameras for each view. This can be an issue when calibrating a multiocular camera system, as the overlapped field of view shrinks as more cameras are considered. Second, estimating extrinsic parameters using an averaged pose is not an

[1] STEREOCALIBRATE actually calculates the median of pose vectors.

elegant solution and the accuracy heavily depends on the selection of reference camera. The development of improvements over the naive approach is therefore required. Key contributions of the proposed method are:

1. *Automatic pairing* of K cameras to establish $K-1$ stereo couples with maximised use of collected calibration data.

2. *Nonlinear bundle adjustment* in a unified coordinate system that simultaneously optimise all the system parameters to achieve a global consistency.

A.3 Multiocular calibration

Calibrating K cameras with a calibration target placed in N different positions defines a K-camera N-view calibration problem. There are K sets of intrinsic parameters, $K - 1$ sets of extrinsic parameters and N sets of target poses to be estimated. The pose parameters are defined by an arbitrarily selected camera space as the reference coordinate system (therefore we calibrate only $K - 1$ instead of K camera poses, as the transformation from the selected camera to the reference is always the identity).

The selection of the referenced camera, however, can be tricky. Due to occlusion of the calibration target, one has to assume it is only observed partially by a subset of cameras for each view. As the result, the extrinsic parameters could not be derived directly using any single view. In this section we describe a multiocular technique that automatically finds a robust way to assemble complete extrinsic parameters using multiple views.

A.3.1 Calibration graph and parameters initialisation

The extrinsic parameters of a multiocular camera system can be modeled by a *complete directed graph* $G = (V_G, E_G)$. A vertex $v \in V$ in the graph represents a camera, and an edge $e_{ij} \in E$ between vertices v_i, v_j denotes the coordinate transformation from v_i to v_j. Assume we have vertices v_i and v_j with their extrinsic parameters $\mathbf{E}_i = (\mathbf{R}_i, \mathbf{t}_i)$ and $\mathbf{E}_j = (\mathbf{R}_j, \mathbf{t}_j)$, the transformation on edge e_{ij} is therefore $\mathbf{E}_{ij} = (\mathbf{R}_j \mathbf{R}_i^{-1}, \mathbf{t}_j - \mathbf{R}_i^{-1} \mathbf{t}_i)$.

Explicitly estimating all $\frac{K(K-1)}{2}$ transformations $\mathbf{E}_{ij}$ is not necessary (and could be impossible due to view occlusion) for the recovery of extrinsic parameters $\mathbf{E}_i, 1 \le i \le K$. As the transformation of coordinate system in Euclidean space is transitive, one can derive $\mathbf{E}_{ij}$ via an agent camera v_k by concatenating $\mathbf{E}_{ik}$ to $\mathbf{E}_{kj}$. Calibrating edges $E_T \subset E_G$ in a spanning tree $T = (V_T, E_T)$ of G is therefore sufficient to recover all the extrinsic parameters.

Costs on the edges have to be defined to form a spanning tree. In practice, different formations of spanning tree lead to numerically distinguishing parameters due to data imperfections. To ensure robustness, we define a cost function subject to the number of shared views between cameras, which assigning to each edge a cost $c_{ij} = \frac{1}{\gamma_{ij}}$ where

$$\gamma_{ij} = \sum_{1 \le r \le N} b_r(i, j) \tag{A.4}$$

and

$$b_r(i, j) = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ see the target in } r\text{-th image} \\ 0, & \text{otherwise} \end{cases} \tag{A.5}$$

This way, cost between strongly connected cameras is low, and the *minimum spanning tree (MST)* yields a more robust initialisation of extrinsic parameters.

Figure A.1 demonstrates an example of calibration graph and its MST formed using the described cost function. In the shown calibration problem the number of shared views between (v_1, v_2), (v_2, v_3), (v_2, v_4) and (v_3, v_4) are 7, 9, 5 and 4 respectively, while pairs (v_1, v_3) and (v_1, v_4) share no views.

We estimate the initial parameters of a multiocular system as follows. First the intrinsic parameters are calibrated independently for each camera. The intrinsics are

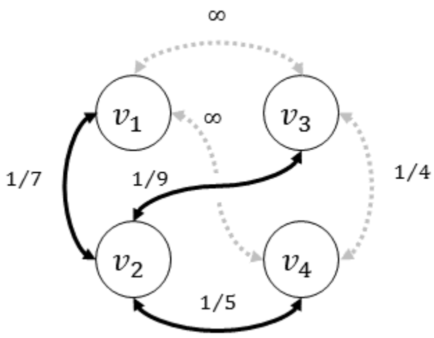


Figure A.1: Example of a calibration graph and its MST.

then used to estimate the transformations pairwise in the MST. The calibration fails if the MST of the calibration graph does not exist (i.e. isolated vertex present). The formation of spanning trees is implemented by means of Prim's algorithm (Boruvka, 1926) in this work.

Two problems remain with the initialised parameters. First, they are solved in a linear manner whereas only algebraical errors are minimised, hence the lack geometric meaning. Second, the estimations are done locally without taking global conditions into account. This could lead to highly biased estimations. The global adjustment as described in next section is therefore suggested to be carried out.

A.3.2 Nonlinear optimisation

The linearly initialised parameters are further tuned using our implementation of the Levenberg-Marquardt (LM) algorithm. The algorithm iteratively searches the optimal parameters $\mathbf{x}$ that minimises a nonlinear objective function $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$ in the sum-of-square form $\Phi(\beta) = \|\mathbf{y} - f(\mathbf{x};\beta)\|^2$, given the observed correspondences $\mathbf{x} \leftrightarrow \mathbf{y}$ and the measurement function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ parametrised by $\beta \in \mathbb{R}^d$.

In iteration k the estimate β_k is updated by $\beta_{k+1} = \beta_k + \Delta\beta_k$ based on the solution of an augmented normal equations:

$$\Delta\beta_k = \left[\mathbf{H} + \lambda \mathrm{diag}(\mathbf{H})\right]^{-1} \varepsilon_k \tag{A.6}$$

with $\mathbf{H} = \mathbf{J}^\top\mathbf{J}$ the Hessian matrix approximated by $\mathbf{J}$ the Jacobian matrix (i.e. $\mathbf{J}_{ij} = \frac{\partial f_i(\mathbf{x}; \beta_k)}{\partial \beta_j}$), $\varepsilon_k = \mathbf{J}^\top[\mathbf{y} - f(\mathbf{x}; \beta_k)]$ the error gradient and λ the damping variable. The differentiation can be achieved using either symbolic or numerical approach, while the numerical differentiation is implemented in this work.

The new estimate β_{k+1} is then assessed by checking $\Phi(\beta_{k+1}) < \Phi(\beta_k)$. For a better result the update is accepted and the damping variable is magnified to $\eta\lambda$ where $\eta > 1$ is a defined multiplier, turning the optimiser toward the gradient descent algorithm. Otherwise we reject the update and decrease the damping variable to $\frac{\lambda}{\eta}$ such that the optimiser will behave more like the Gauss-Newton approach.

Termination criteria indicating convergence of the optimisation process are assessed over time. The error drop and step size are considered good indicators of convergence. In particular, we check the conditions

$$\frac{\Phi(\beta_k) - \Phi(\beta_{k+1})}{\Phi(\beta_k)} < \epsilon \tag{A.7}$$

and

$$\frac{\|\Delta\beta_k\|}{\|\beta_k\|} < \epsilon \tag{A.8}$$

with the machine precision ϵ, which is usually set to a small positive value below which the change is numerically meaningless (Press, Teukolsky, Vetterling & Flannery, 2007). Setting higher ϵ tends to terminate the optimisation prematurely, while having a low ϵ could be a waste of time.

In the context of camera calibration, the 3D-to-2D correspondences of control points are used to instantiate the minimisation problem, f actualises the camera's projection function, and the least square reprojection error (RPE) is attained by the optimal parameters. In Bayesian terms, LM yields the maximum likelihood estimation of the parameters with the highest probability given the observations, if the noises of calibration data are Gaussian.

The parameters to be optimised include K-set intrinsics, $(K-1)$-set extrinsics and N-set target poses, as follows:

$$\beta = (\mu_1, \mu_2, ..., \mu_K, \xi_1, \xi_2, ..., \xi_{K-1}, \rho_1, \rho_2, ..., \rho_N) \tag{A.9}$$

where each intrinsic vector $\mu = (f_x, f_y, x_c, y_c, \kappa_1, \kappa_2, \kappa_3, p_1, p_2)$ encodes camera's intrinsic parameters, while ξ_i and ρ_i respectively describe the extrinsic parameters and target poses in the 6-vector form $(a_x, a_y, a_z, t_x, t_y, t_z)$ with first 3 components (a_x, a_y, a_z) define the angle-axis representation of rotation, and last 3 components (t_x, t_y, t_z) represent the translation. We therefore have $d = 9K + 6(K-1) + 6N$ the dimension of parameter space. Figure A.2 visualises the Hessian matrix in a real case with respect to β of the described vector form.

A.4 Experiments

We applied the developed technique to calibrate a trinocular camera system, and the results are compared with the direct extension of the stereo calibration that comes with `OpenCV`, which estimates parameters pairwise. A chessboard has been placed in 38 poses and simultaneously imaged by 3 cameras. Figure A.3 shows the positions of the calibration target observed by the first camera.

The global optimisation runs for at most 30 iterations, and we set machine epsilon $\epsilon = 10^{-8}$ for convergence check. The root-mean-square errors (RMSEs) of reprojected control points are plotted in Fig. A.4. Significant error drops are observed through

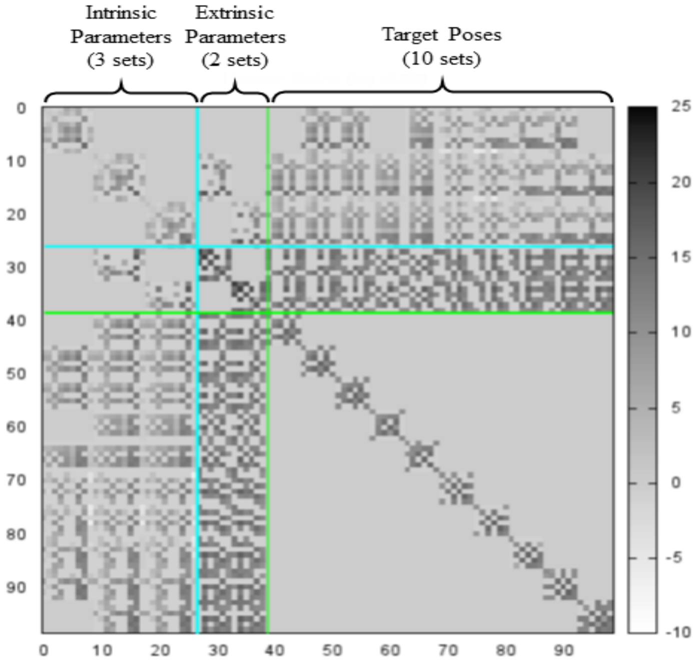


Figure A.2: Hessian matrix (in logarithmic scale) of a 3-camera 10-view calibration problem.

iteration 4 to 8. In the first four iterations the improvement is not significant. A possible explanation is that, the linearly initialised parameters are very close to a local optimum. In this case, an insufficient termination criteria, say $\epsilon = 10^{-4}$, could lead to an early stop of the optimisation process.

In Fig. A.5 the control points are reprojected using parameters obtained and locally optimised by the `OpenCV` calibration subroutine, without global adjustment. Compared to Fig. A.6 which shows the reprojections optimised according to Section A.3.2, an improvement is indicated by the drops of RMSE from 0.3 to 0.07 pixels for the second and the third cameras respectively, despite the error drop is not significant for the first camera.

Image rectification is applied to study the accuracy of calibrated parameters. The rectified trinocular views are shown in Fig. A.7 and Fig. A.8, respectively of pairwise and globally optimised parameters. As can be seen, the epipolar constraint is well

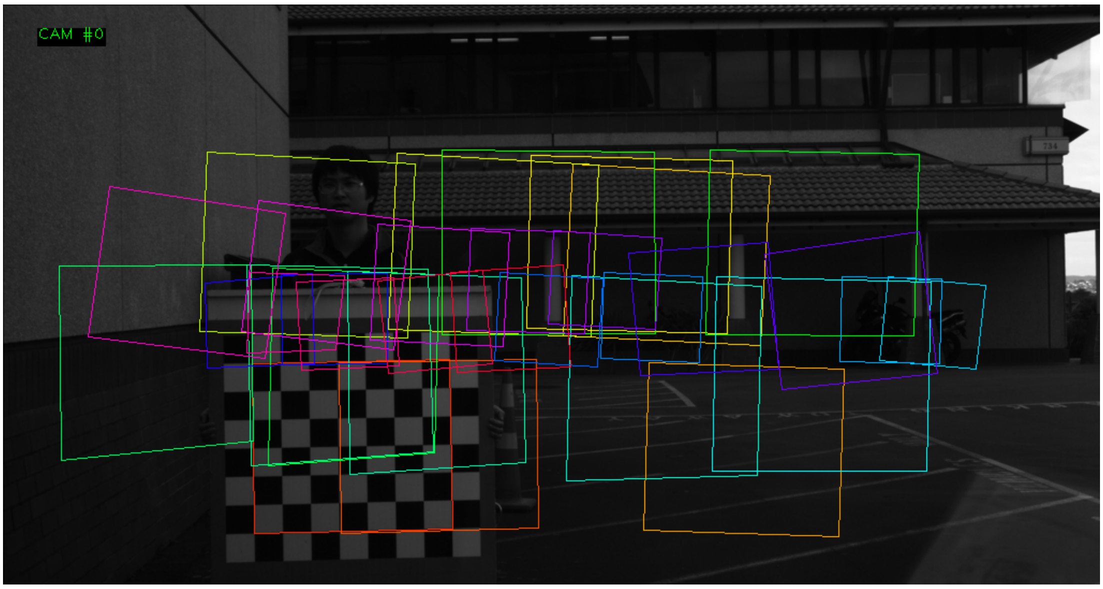


Figure A.3: Coverage of 38 placements of the calibration target.

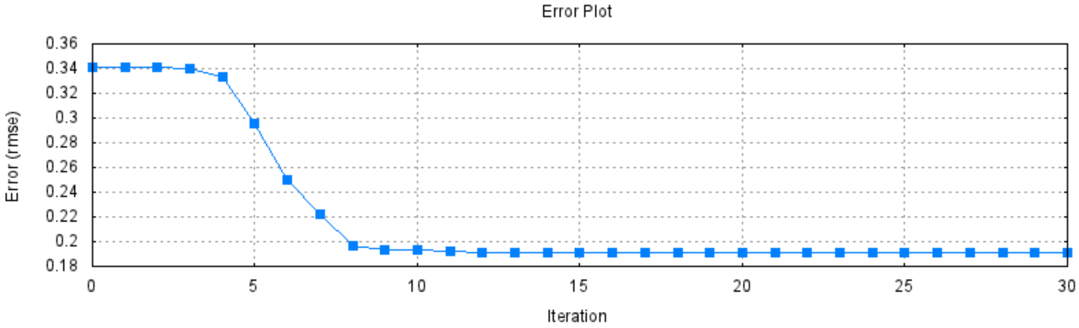


Figure A.4: Error plot through the global optimisation process.

preserved near the marked control points in both cases. However, in the remaining (e.g. the top margin of the image) significant misalignment is obvious in Fig. A.7. This demonstrates the generalisation error due to the bias of locally optimised parameters.

A.5 Summary

A multiocular camera calibration tool has been demonstrated. The proposed method automatically establishes robust calibration pairs by solving the MST problem. The initialised parameters are further optimised in global scale using our implemented LM

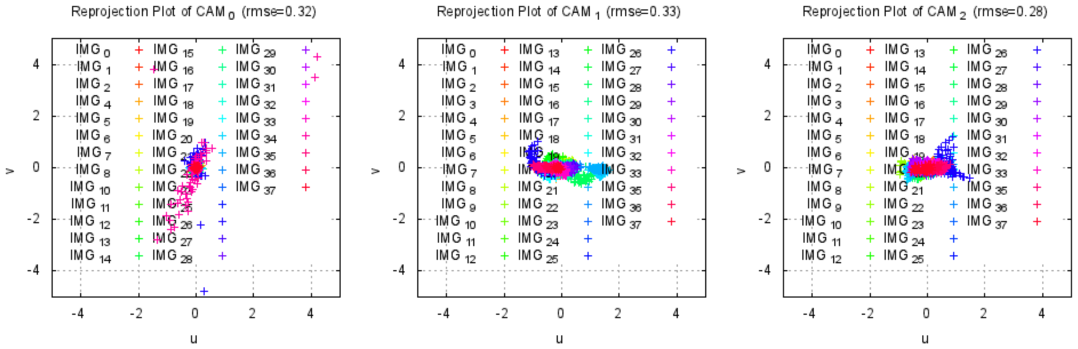


Figure A.5: Reprojection error of control points using pairwise optimised parameters without global bundle adjustment.

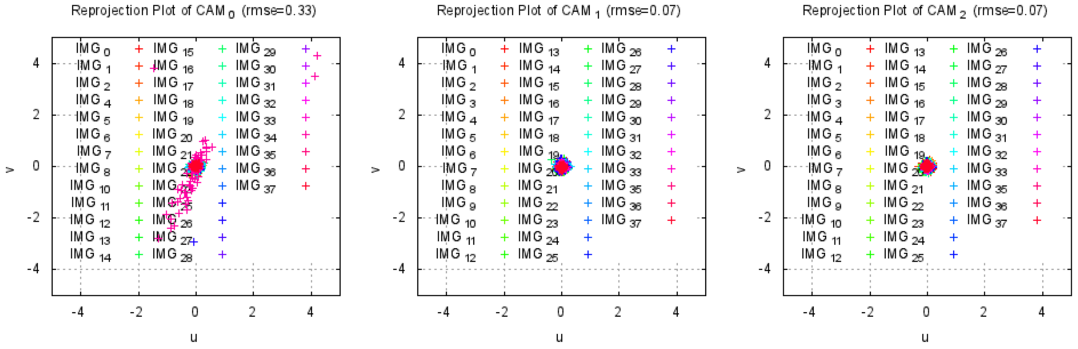


Figure A.6: Control points reprojected using globally optimised parameters.

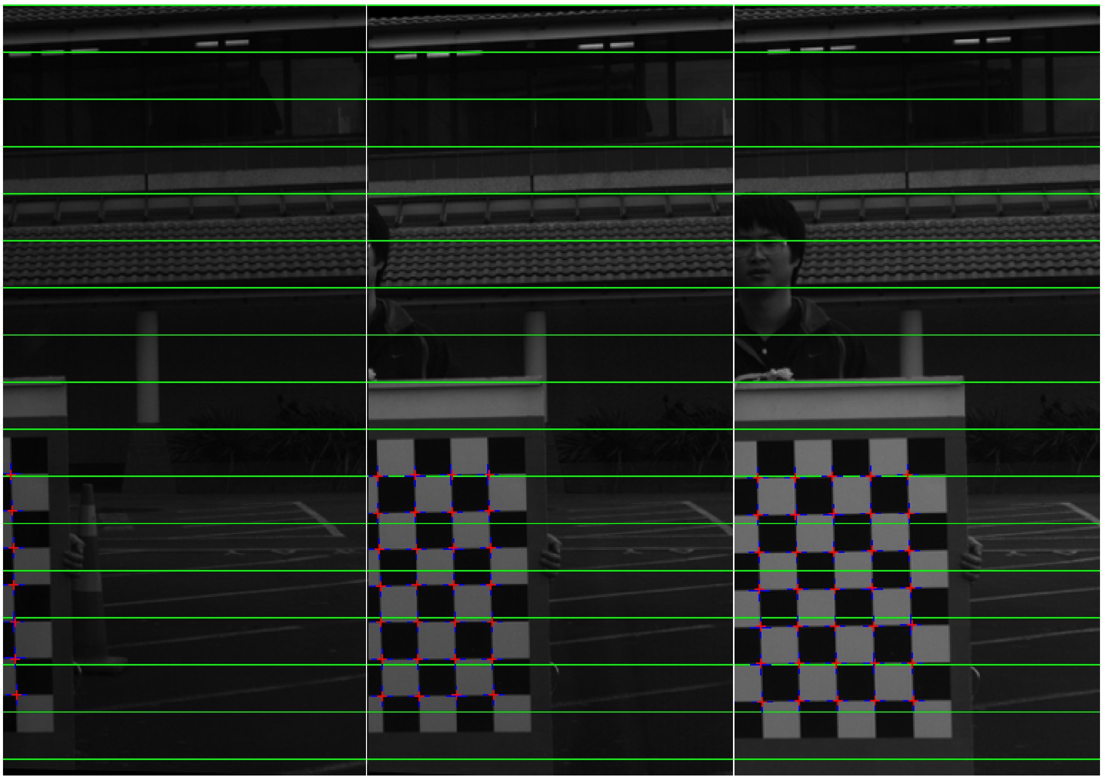


Figure A.7: Views rectified using pairwise optimised parameters without global bundle adjustment. Note the fluorescent lamps on the left-top of the middle and right images are not vertically aligned to the epipolar line.

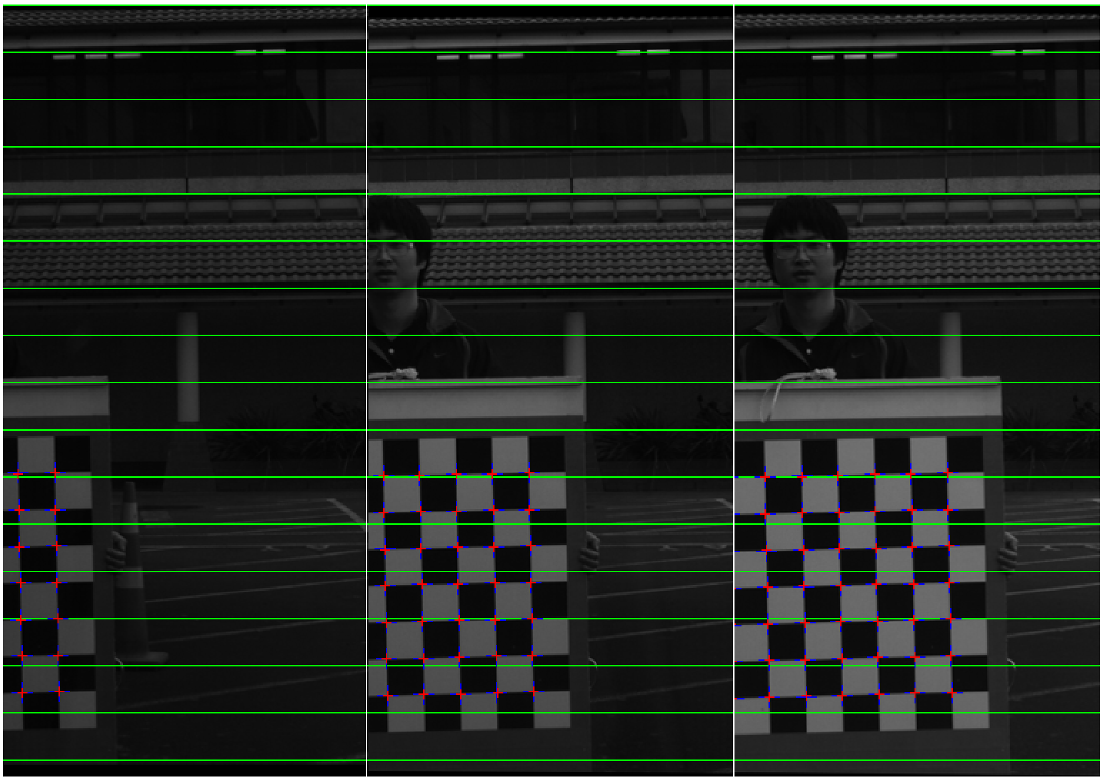


Figure A.8: Rectified views using globally optimised parameters.

algorithm. As supported by the experimental results, the proposed method is able to attain globally consistent parameters. Compared to the direct extension of stereo calibration algorithm, an improvement of 44% in RPE is attainable.

Appendix B

Comparative Study on Image Features

B.1 Image features

In this section we provide brief descriptions of the types of image features considered in this thesis.

B.1.1 SIFT

The SIFT (scale-invariant feature transform) algorithm, proposed by David Lowe in 1999 (Lowe, 2004), is perhaps one of the earliest work on providing a comprehensive keypoint detection and descriptor extraction technique.

To formulate a scale invariant representation of image features, SIFT builds a multi-resolution pyramid over the input image and applies *difference of Gaussians* (DoG) operators to locate local extrema in the scale space. The locality is defined by a $3 \times 3 \times 3$ window. These non-edge extrema, presenting high local contrast, are then identified as the keypoints.

The image gradients of a 16×16 window, centred at each keypoint, are then computed and grouped into 4×4 subregions. The direction of gradients within the same subregion are then quantised to conclude an eight-bins histogram. By gathering all the bins from all sixteen histograms in the window, one obtains an 128-element SIFT descriptor vector.

B.1.2 SURF

Six years after the advent of SIFT, Herbert Bay et. al. proposed an alternative image feature detection and extraction algorithm known as SURF (speeded-up robust features), which was claimed to be faster and more robust than SIFT (Bay et al., 2006). Instead of the DoG operators, the SURF algorithm chooses a box-filter-approximated second-order derivative computation to locate extrema in the scale space. These Haar-like operators can be efficiently implemented over a computed integral image. This leads to a speed-up of SURF compared to SIFT.

The robustness of SURF features comes from the identification of patch direction before extracting its feature vector. The direction of a keypoint is decided based on Gaussian-weighted responses of pixels in a circular neighbourhood with respect to horizontal and vertical Haar wavelets. These responses are transformed into polar coordinates, and partitioned based on a predefined angular resolution. The transformed response vectors in the same partition are then summed up, and the angle of the longest vector sum over all partitions is chosen as the direction of the patch. Based on the direction, the patch is rotated and a Gaussian-weighted discrete sampling on pixel responses to the Haar wavelets is performed to yield a real-vector descriptor.

B.1.3 ORB

In 2011, Ethan Rublee et. al. proposed the ORB (Oriented FAST and Rotated BRIEF) features as an alternative to SIFT and SURF (Rublee, Rabaud, Konolige & Bradski, 2011). As its self-explanatory name, the ORB algorithm combines an enhanced FAST (features from accelerated segment test) (Rosten, Porter & Drummond, 2010) technique to locate keypoints, with a direction-normalised BRIEF (binary robust independent elementary) (Calonder, Lepetit, Strecha & Fua, 2010) descriptor extraction process. To achieve scale invariance, the FAST algorithm is applied repeatedly to each layer of a scale pyramid. The cornerness of detected FAST features is tested using a Harris measurement (Harris & Stephens, 1988) and non-corner keypoints are excluded.

The BRIEF algorithm produces an n-bit string from local binary tests of n predefined pixel pairs within a patch. This vector representation would be highly unstable against rotation. To address this issue, ORB adopts a rotation-aware variant of BRIEF. For the located keypoints, it first finds patch centroids by means of image moments (Klette, 2014). The direction of a vector, connecting a keypoint's centre to its patch's centroid, then decides the direction of the keypoint. The binary test pattern is then rotated by the direction of a patch before binary descriptor extraction, allowing a feature to be represented in a rotation-invariant (i.e. isotropic) form.

B.1.4 A-KAZE

More recently, Pablo F. Alcantarilla el. al. developed an edge-preserving non-linear filtering strategy to locate image features (Alcantarilla, Bartoli & Davison, 2012). The filtering is based on the image diffusion equation:

$$\frac{\partial I}{\partial t}(u,v,t) = \text{div}\left[c(u,v,t) \cdot \nabla I(u,v)\right] \tag{B.1}$$

where div and ∇ are, respectively, the divergence and the gradient operator; I is an intensity image, $t > 0$ is the scale variable, and c a conductivity function.

The diffusion of an image formulates an iterative way to find the filtered, yet edge-preserved evolution of the original image, by choosing the conductivity function carefully. A good choice of the conductivity is as follows:

$$c(u, v, t) = \left[1 + \left(\frac{\nabla I_\sigma(u, v, t)}{k} \right)^2 \right]^{-1} \tag{B.2}$$

where I_σ is a Gaussian-smoothed version of image I. In the follow-up work the *fast explicit diffusion* (FED) technique is used to solve the diffusion equation efficiently, contributing to an accelerated variation of the KAZE algorithm, proposed by the authors in 2012 (Alcantarilla, Nuevo & Bartoli, 2013). The improved feature detection and extraction technique is known as A-KAZE.

Keypoints are located by finding the extrema of the second-order derivatives of the image over the non-linear multi-scale pyramid built from the principle of image diffusion. A-KAZE deploys a technique similar to SURF to estimate the direction of a patch. A modified *local difference binary* (LDB) representation of the rotation-compensated patch is then extracted as its binary descriptor.

B.2 Experiments

We present results for a test sequence selected from the KITTI benchmark datasets (Geiger et al., 2013) to evaluate the image features. The sequence presents a complex street scenario. Moving bicyclists, vehicles, and walking pedestrians are present in the scene. The test vehicle travelled 300 metres and captured 389 frames. We selected gray-level data of the left camera to perform monocular visual odometry. Data of the remaining three cameras are not used.

We use feature detection and extraction subroutines shipped with `OpenCV 3.1`. These subroutines are CPU-only implementations. The default parameters provided by the library are used, with only one exception: We increased the number of detectable features to $3,000$ for ORB, as its default setting to 500 features is insufficient to produce

Table B.1: Comparison of feature extraction

	SIFT	SURF	ORB	A-KAZE
# Features	765,436	1,140,410	1,017,350	603,246
Time spent	98 sec	50 sec	10 sec	43 sec
Disk storage	391 MB	305 MB	54 MB	49 MB

good results, compared to the other tested features. The obtained statistics of the feature extraction process is reported in Table B.1.

For each tested feature, we repeat the visual odometry process 48 times on a quad-core Intel Core i7 3.2 GHz laptop. Motion drifts are then evaluated using ground truth available for these KITTI data (derived from GPS/IMU readings). To allow the ego-motion estimation to be evaluated in a consistent scale, the inertial data of the first and the second frame are used to bootstrap the monocular visual odometry process.

The mean drift of each run is calculated individually; the results of all 192 runs are plotted together in Fig. B.1. The plot shows that SIFT and SURF features achieve similar levels of accuracy, while the error distribution of the runs using SURF features is slightly less dispersed, compared to SIFT. The ORB features, on the other hand, show inconsistent performance with a wide range of drift errors. The A-KAZE features yield intermediate performance in-between the cases of SURF and ORB.

The best run of each trial set is selected to render a detailed view of how the different image features perform through the visual odometry process. The inter-frame and overall drift errors of these runs are plotted in Fig. B.2. Initially, all the tested features shared a similar performance as the vehicle is slowly accelerating. The drift of the ego-motion, estimated from ORB features, however, grows faster than other cases, as the vehicle reaches a moderate speed. At the end of the sequence, the drift error follows the pattern “ORB > A-KAZE > SIFT > SURF”. It is also found that ORB-based visual odometry is less stable, as significant inter-frame drift errors occur

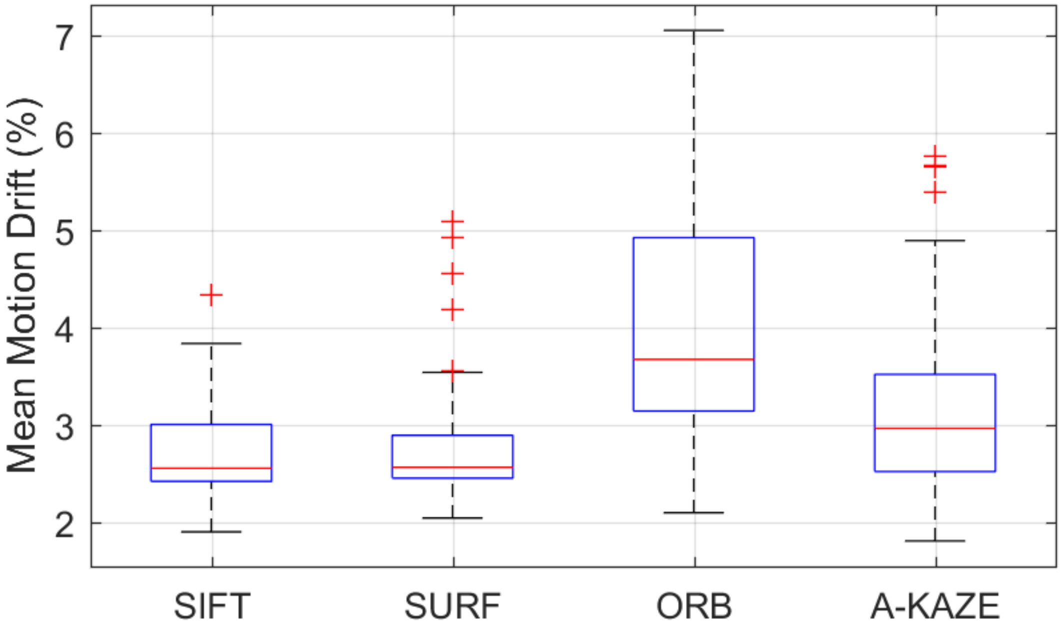


Figure B.1: Box plot of tested features. For each feature, the quartiles are concluded from 48 individual visual-odometry processes. The first and the third quartiles are, respectively, denoted by the bottoms and tops of a box, and the second quartile (median) is marked by the red line in the box.

more frequently.

The segmented motion errors, in terms of translation and rotation components of the estimated ego-motion, are also computed, with respect to various travel distances.

The travel distance is not measured only from the beginning of the sequence; applying the segmented error analysis of KITTI, we consider segments that begin from an arbitrary frame k through to frame $k+n$, where $n > 0$. The actual length of a segment is taken into account when calculating the error at image I being in a temporal interval $[l_p, l_q)$, with temporal order $l_p \leq l < l_q$.

We divide the length of the sequence into 10 equally spaced segments for plotting. The results are depicted in Fig. B.3. It shows that, in the translation component, the A-KAZE features perform best for short movement within 150 metres, and the SURF features achieve the lowest overall error of 1% as the vehicle reaches the end of the sequence. In the rotational component, it presents a clear trend that the SIFT and SURF

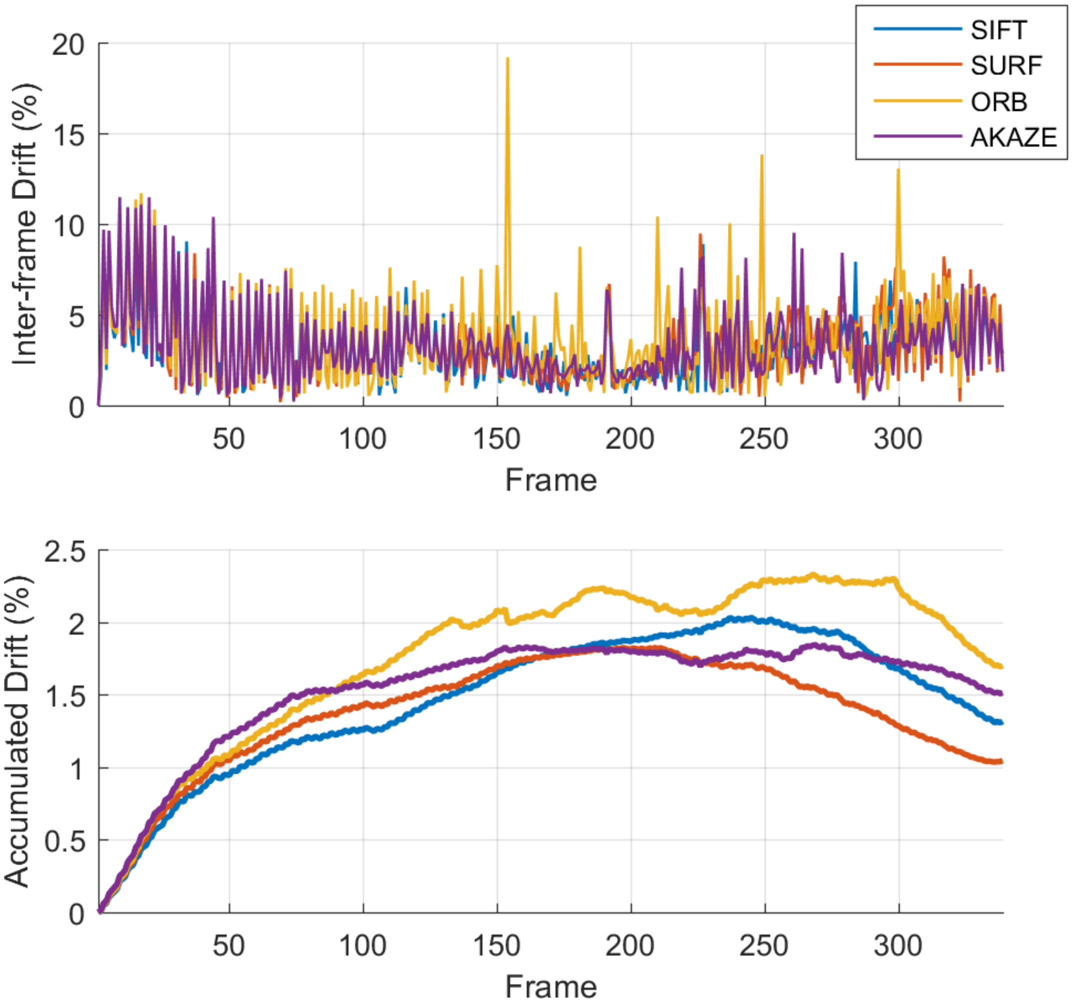


Figure B.2: Inter-frame (top) and accumulated (bottom) drift errors of the best cases of SIFT, SURF, ORB, and A-KAZE features.

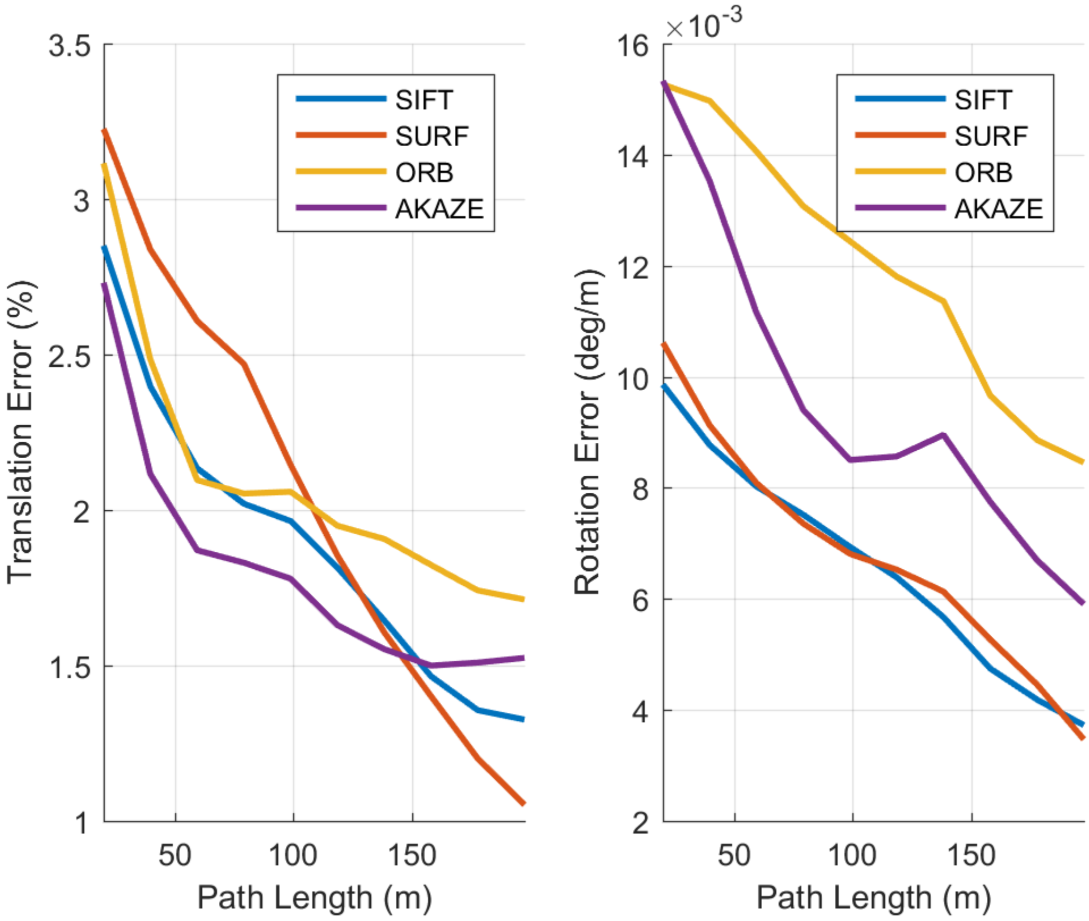


Figure B.3: Segmented motion error analysis on the translational (left) and rotational (right) components.

cases converge similarly to the lowest error, while the ORB features achieve two-times worse accuracy, and the A-KAZE features show an intermediate error.

References

Alcantarilla, P. F., Bartoli, A. & Davison, A. J. (2012). Kaze features. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato & C. Schmid (Eds.), *Computer vision – eccv 2012: 12th european conference on computer vision, florence, italy, october 7-13, 2012, proceedings, part vi* (pp. 214–227). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from `https://doi.org/10.1007/978-3-642-33783-3_16` doi: 10.1007/978-3-642-33783-3_16

Alcantarilla, P. F., Nuevo, J. & Bartoli, A. (2013). Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *Bmvc.*

Badino, H., Franke, U. & Pfeiffer, D. (2009). The stixel world - a compact medium level representation of the 3d-world. In *Proceedings of the 31st dagm symposium on pattern recognition* (pp. 51–60). Berlin, Heidelberg: Springer-Verlag. Retrieved from `http://dx.doi.org/10.1007/978-3-642-03798-6_6` doi: 10.1007/978-3-642-03798-6_6

Badino, H., Yamamoto, A. & Kanade, T. (2013, Dec). Visual odometry by multi-frame feature integration. In *2013 ieee international conference on computer vision workshops* (p. 222-229). doi: 10.1109/ICCVW.2013.37

Baker, S. & Matthews, I. (2004, Feb). Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, *56*(3), 221–255. Retrieved from `http://dx.doi.org/10.1023/B:VISI.0000011205.11775.fd` doi: 10.1023/B:VISI.0000011205.11775.fd

Bay, H., Tuytelaars, T. & Van Gool, L. (2006). SURF: Speeded up robust features. In A. Leonardis, H. Bischof & A. Pinz (Eds.), *Computer vision – eccv 2006: 9th european conference on computer vision, graz, austria, may 7-13, 2006. proceedings, part i* (pp. 404–417). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from `https://doi.org/10.1007/11744023_32` doi: 10.1007/11744023_32

Beardsley, P., Zisserman, A. & Murray, D. (1997, 01 Jun). Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, *23*(3), 235–259. Retrieved from `https://doi.org/10.1023/A:1007923216416` doi: 10.1023/A:1007923216416

Besl, P. J. & McKay, N. D. (1992, Feb). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239-256. doi: 10.1109/34.121791

Boruvka, O. (1926). O Jistém Problému Minimálním (About a Certain Minimal

Problem) (in Czech, German summary). *Práce Mor. Prírodoved. Spol. v Brne III*, *3*.

Bouguet, J. Y. (2015). *Camera calibration toolbox for Matlab.* Retrieved from http://www.vision.caltech.edu/bouguetj/calib_doc/.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., ... Leonard, J. (2016). Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332.

Calonder, M., Lepetit, V., Strecha, C. & Fua, P. (2010). Brief: Binary robust independent elementary features. In *Proceedings of the 11th european conference on computer vision: Part iv* (pp. 778–792). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=1888089.1888148

Chien, H.-J., Geng, H., Chen, C.-Y. & Klette, R. (2016). Multi-frame feature integration for multi-camera visual odometry. In *Revised selected papers of the 7th pacific-rim symposium on image and video technology - volume 9431* (pp. 27–37). New York, NY, USA: Springer-Verlag New York, Inc. Retrieved from http://dx.doi.org/10.1007/978-3-319-29451-3_3 doi: 10.1007/978-3-319-29451-3_3

Chien, H.-J., Geng, H. & Klette, R. (2014). Improved visual odometry based on transitivity error in disparity space: A third-eye approach. In *Proceedings of the 29th international conference on image and vision computing new zealand* (pp. 72–77). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2683405.2683427 doi: 10.1145/2683405.2683427

Chien, H.-J., Geng, H. & Klette, R. (2015). Bundle adjustment with implicit structure modeling using a direct linear transform. In G. Azzopardi & N. Petkov (Eds.), *Computer analysis of images and patterns: 16th international conference, caip 2015, valletta, malta, september 2-4, 2015 proceedings, part i* (pp. 411–422). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-23192-1_34 doi: 10.1007/978-3-319-23192-1_34

Chien, H.-J. & Klette, R. (2017). Regularised energy model for robust monocular ego-motion estimation. In *Proceedings of the 12th international joint conference on computer vision, imaging and computer graphics theory and applications - volume 6: Visapp, (visigrapp 2017)* (pp. 361–368). SciTePress. doi: 10.5220/0006100303610368

Chien, H.-J., Klette, R., Schneider, N. & Franke, U. (2016, Dec). Visual odometry driven online calibration for monocular lidar-camera systems. In *2016 23rd international conference on pattern recognition (icpr)* (p. 2848-2853). doi: 10.1109/ICPR.2016.7900068

Choi, J., Kim, H., Oh, T. H. & Kweon, I. S. (2014, Oct). Balanced optical flow refinement by bidirectional constraint. In *2014 ieee international conference on image processing (icip)* (p. 5477-5481). doi: 10.1109/ICIP.2014.7026108

Choi, S., Kim, T. & Yu, W. (2009). Performance evaluation of RANSAC family. In *Bmvc* (pp. 1–12). Retrieved from http://dx.doi.org/10.5244/c.23.81

doi: 10.5244/c.23.81

Chojnacki, W. & Brooks, M. J. (2003, Sept). Revisiting hartley's normalized eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(9), 1172-1177. doi: 10.1109/TPAMI.2003.1227992

Ci, W. & Huang, Y. (2016). A robust method for ego-motion estimation in urban environment using stereo camera. *Sensors*, *16*(10). Retrieved from http://www.mdpi.com/1424-8220/16/10/1704 doi: 10.3390/s16101704

Civera, J., Davison, A. J. & Montiel, J. M. M. (2008, Oct). Inverse depth parametrization for monocular slam. *IEEE Transactions on Robotics*, *24*(5), 932-945. doi: 10.1109/TRO.2008.2003276

Durrant-Whyte, H. & Bailey, T. (2006, 05 Jun). Simultaneous Localisation and Mapping (SLAM): Part i the essential algorithms. *Robotics & Automation Magazine, IEEE*, *13*(2), 99–110. Retrieved from http://dx.doi.org/10.1109/mra.2006.1638022 doi: 10.1109/mra.2006.1638022

Engel, J., Schöps, T. & Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In D. Fleet, T. Pajdla, B. Schiele & T. Tuytelaars (Eds.), *Computer vision – eccv 2014: 13th european conference, zurich, switzerland, september 6-12, 2014, proceedings, part ii* (pp. 834–849). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-10605-2_54 doi: 10.1007/978-3-319-10605-2_54

Engel, J., Sturm, J. & Cremers, D. (2013, Dec). *Semi-dense visual odometry for a monocular camera.* doi: 10.1109/ICCV.2013.183

Engels, C., Stewénius, H. & Nistér, D. (2006). Bundle adjustment rules. In *In photogrammetric computer vision.*

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Proceedings of the 13th scandinavian conference on image analysis* (pp. 363–370). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=1763974.1764031

Fischler, M. A. & Bolles, R. C. (1981, June). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395. Retrieved from http://doi.acm.org/10.1145/358669.358692 doi: 10.1145/358669.358692

Forster, C., Pizzoli, M. & Scaramuzza, D. (2014, May). *Svo: Fast semi-direct monocular visual odometry.* doi: 10.1109/ICRA.2014.6906584

Fraundorfer, F. & Scaramuzza, D. (2012, 16 Feb). Visual odometry: Part ii - matching, robustness, and applications. *IEEE Robotics & Automation Magazine*, *19*(1), 78-90.

Galvez-López, D. & Tardos, J. D. (2012, Oct). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, *28*(5), 1188-1197. doi: 10.1109/TRO.2012.2197158

Gao, X.-S., Hou, X.-R., Tang, J. & Cheng, H.-F. (2003, Aug). Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(8), 930-943. doi: 10.1109/TPAMI.2003.1217599

Gehrig, S. K. & Stein, F. J. (1999). Dead reckoning and cartography using stereo vision for an autonomous car. In *Proceedings 1999 ieee/rsj international conference on intelligent robots and systems. human and environment friendly robots with high intelligence and emotional quotients* (Vol. 3, pp. 1507–1512). doi: 10.1109/IROS.1999.811692

Geiger, A., Lenz, P., Stiller, C. & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR).*

Geiger, A., Ziegler, J. & Stiller, C. (2011). *Stereoscan: Dense 3d reconstruction in real-time.*

Harris, C. & Stephens, M. (1988). A combined corner and edge detector. In *In proc. of fourth alvey vision conference* (pp. 147–151).

Hartley, R. I. (1997, Jun). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(6), 580-593. doi: 10.1109/34.601246

Hartley, R. I. & Sturm, P. (1995). Triangulation. In V. Hlaváč & R. Šára (Eds.), *Computer analysis of images and patterns: 6th international conference, caip'95 prague, czech republic, september 6–8, 1995 proceedings* (pp. 190–197). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from `https://doi.org/10.1007/3-540-60268-2_296` doi: 10.1007/3-540-60268-2_296

Hartley, R. I. & Zisserman, A. (2004). *Multiple view geometry in computer vision* (Second ed.). Cambridge University Press.

Hirschmüller, H. (2008, Feb). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(2), 328-341. doi: 10.1109/TPAMI.2007.1166

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, *4*(4), 629–642.

Horn, B. K. P. & Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, *17*(1), 185 - 203. Retrieved from `http://www.sciencedirect.com/science/article/pii/0004370281900242` doi: http://dx.doi.org/10.1016/0004-3702(81)90024-2

Hu, J., You, S. & Neumann, U. (2003, Nov). Approaches to large-scale urban modeling. *IEEE Comput. Graph. Appl.*, *23*(6), 62–69. Retrieved from `http://dx.doi.org/10.1109/MCG.2003.1242383` doi: 10.1109/MCG.2003.1242383

Irani, M. & Anandan, P. (1999). All about direct methods. In *Proceedings of iccv workshop vision algorithms: Theory practice* (p. 267-277).

Kanade, T. & Morris, D. D. (1998). Factorization methods for structure from motion. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *356*(1740), 1153–1173. Retrieved from `http://rsta.royalsocietypublishing.org/content/356/1740/1153` doi: 10.1098/rsta.1998.0215

Karlsson, N., di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P. & Munich, M. E. (2005, April). The vslam algorithm for robust localization and mapping. In *Proceedings of the 2005 ieee international conference on robotics and automation* (p. 24-29). doi: 10.1109/ROBOT.2005.1570091

Kazhdan, M. & Hoppe, H. (2013, July). Screened poisson surface reconstruction. *ACM Trans. Graph.*, *32*(3), 29:1–29:13. Retrieved from http://doi.acm.org/10.1145/2487228.2487237 doi: 10.1145/2487228.2487237

Kerl, C., Sturm, J. & Cremers, D. (2013). Dense visual slam for rgb-d cameras. In *Proc. of the int. conf. on intelligent robot systems (iros).*

Kim, W. S., Ansar, A. I., Steele, R. D. & Steinke, R. C. (2005, Oct). Performance analysis and validation of a stereo vision system. In *2005 ieee international conference on systems, man and cybernetics* (Vol. 2, p. 1409-1416). doi: 10.1109/ICSMC.2005.1571344

Kitt, B., Geiger, A. & Lategahn, H. (2010, June). Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *2010 ieee intelligent vehicles symposium* (p. 486-492). doi: 10.1109/IVS.2010.5548123

Klein, G. & Murray, D. (2007, Nov). Parallel tracking and mapping for small ar workspaces. In *2007 6th ieee and acm international symposium on mixed and augmented reality* (p. 225-234). doi: 10.1109/ISMAR.2007.4538852

Klette, R. (2014). *Concise computer vision: An introduction into theory and algorithms*. Springer Publishing Company, Incorporated.

Knuth, D. E., Larrabee, T. & Roberts, P. M. (1989). *Mathematical writing*. Washington, DC, USA: Mathematical Association of America.

Konolige, K. (2010). Sparse sparse bundle adjustment. In *Proceedings of the british machine vision conference* (pp. 102.1–102.11). BMVA Press. (doi:10.5244/C.24.102)

Lepetit, V., Moreno-Noguer, F. & Fua, P. (2009). Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, *81*(2).

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, *2*(2), 164-168. Retrieved from http://www.jstor.org/stable/43633451

Li, H. & Hartley, R. (2006, Aug). *Five-point motion estimation made easy* (Vol. 1). doi: 10.1109/ICPR.2006.579

Lindstrom, P. (2009, Dec). Triangulation made easy.. Retrieved from http://www.osti.gov/scitech/servlets/purl/983384

Longuet-Higgins, H. C. (1981, 10 Sep). A computer algorithm for reconstructing a scene from two projections. *Nature*, *293*(5828), 133–135. Retrieved from http://dx.doi.org/10.1038/293133a0 doi: 10.1038/293133a0

Lourakis, M. I. A. & Argyros, A. A. (2009, Mar). Sba: A software package for generic sparse bundle adjustment. *ACM Trans. Math. Softw.*, *36*(1), 1–30. Retrieved from http://doi.acm.org/10.1145/1486525.1486527 doi: 10.1145/1486525.1486527

Lowe, D. G. (2004, Nov). Distinctive image features from scale-invariant keypoints. In (Vol. 60, pp. 91–110). Hingham, MA, USA: Kluwer Academic Publishers. Retrieved from http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94 doi: 10.1023/B:VISI.0000029664.99615.94

Lowry, S., Sunderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P. & Milford,

M. J. (2016, Feb). Visual place recognition: A survey. *Trans. Rob.*, *32*(1), 1–19. Retrieved from https://doi.org/10.1109/TRO.2015.2496823 doi: 10.1109/TRO.2015.2496823

Lucas, B. D. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on artificial intelligence - volume 2* (pp. 674–679). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=1623264.1623280

Luong, Q.-T. & Faugeras, O. (1997, 01 Mar). Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, *22*(3), 261–289. Retrieved from https://doi.org/10.1023/A:1007982716991 doi: 10.1023/A:1007982716991

Maimone, M., Cheng, Y. & Matthies, L. (2007, March). Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics, Special Issue on Space Robotics*, *24*(3), 169–186.

Min, H., Xu, Z., Li, X., Zhang, L. & Zhao, X. (2016). An improved method of vehicle ego-motion estimation based on stereo vision. In *Transportation infrastructure and materials* (pp. 442–452).

Mitra, N. J., Nguyen, A. & Guibas, L. (2004). Estimating surface normals in noisy point cloud data. In *special issue of international journal of computational geometry and applications* (Vol. 14, pp. 261–276).

Moosmann, F. & Stiller, C. (2011, June). Velodyne slam. In *2011 ieee intelligent vehicles symposium (iv)* (pp. 393–398). doi: 10.1109/IVS.2011.5940396

Morris, J., Haeusler, R., Jiang, R., Jawed, K., Kalarot, R., Khan, T., ... Klette, R. (2009, Nov). Current work in the .enpeda.. project. In *2009 24th international conference image and vision computing new zealand* (p. 130-135). doi: 10.1109/IVCNZ.2009.5378425

Muja, M. & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *In visapp international conference on computer vision theory and applications* (pp. 331–340).

Mur-Artal, R., Montiel, J. M. M. & Juan Tardós, D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *CoRR*, *abs/1502.00956*. Retrieved from http://arxiv.org/abs/1502.00956

Nedevschi, S., Bota, S. & Tomiuc, C. (2009, Sept). Stereo-based pedestrian detection for collision-avoidance applications. *IEEE Transactions on Intelligent Transportation Systems*, *10*(3), 380–391. doi: 10.1109/TITS.2008.2012373

Newcombe, R. A., Lovegrove, S. J. & Davison, A. J. (2011, Nov). *Dtam: Dense tracking and mapping in real-time.* doi: 10.1109/ICCV.2011.6126513

Nistér, D. (2003, Oct). Preemptive ransac for live structure and motion estimation. In *Proceedings ninth ieee international conference on computer vision* (Vol. 1, pp. 199–206). doi: 10.1109/ICCV.2003.1238341

Nistér, D. (2004, June). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(6), 756–770. doi: 10.1109/TPAMI.2004.17

Nistér, D., Naroditsky, O. & Bergen, J. (2004, June). Visual odometry. In *Proceedings of the 2004 ieee computer society conference on computer vision and pattern recognition, 2004.* (Vol. 1, pp. 652–659). doi: 10.1109/CVPR.2004.1315094

N. Savinov, L. L., C. Häne & Pollefeys, M. (2016). Semantic 3d reconstruction with continuous regularization and ray potentials using a visibility consistency constraint. In *Proc. ieee int. conf. on computer vision and pattern recognition.*

Olson, C. F., Matthies, L. H., Schoppers, H. & Maimone, M. W. (2000). Robust stereo ego-motion for long distance navigation. In *Proceedings ieee conference on computer vision and pattern recognition. cvpr 2000 (cat. no.pr00662)* (Vol. 2, pp. 453–458). doi: 10.1109/CVPR.2000.854879

The opencv reference manual: Camera calibration and 3d reconstruction (3.3.0.0 ed.) [Computer software manual]. (2017, Aug).

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing* (3rd ed.). New York, NY, USA: Cambridge University Press.

Rosten, E. & Drummond, T. (2005, Oct). Fusing points and lines for high performance tracking. In *Tenth ieee international conference on computer vision (iccv'05) volume 1* (Vol. 2, pp. 1508–1515). doi: 10.1109/ICCV.2005.104

Rosten, E., Porter, R. & Drummond, T. (2010, Jan). Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(1), 105-119. doi: 10.1109/TPAMI.2008.275

Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the 2011 international conference on computer vision* (pp. 2564–2571). Washington, DC, USA: IEEE Computer Society. Retrieved from `http://dx.doi.org/10.1109/ICCV.2011.6126544` doi: 10.1109/ICCV.2011.6126544

Rusu, R. B., Blodow, N. & Beetz, M. (2009, May). Fast point feature histograms (fpfh) for 3d registration. In *2009 ieee international conference on robotics and automation* (pp. 3212–3217). doi: 10.1109/ROBOT.2009.5152473

Rusu, R. B., Blodow, N., Marton, Z. C. & Beetz, M. (2008, Sep.). Aligning point cloud views using persistent feature histograms. In *2008 ieee/rsj international conference on intelligent robots and systems* (p. 3384-3391). doi: 10.1109/IROS.2008.4650967

Scaramuzza, D. & Fraundorfer, F. (2011). Visual odometry: Part i - the first 30 years and fundamentals. *IEEE Robotics & Automation Magazine*, *18*(4), 80-92.

Scharstein, D. & Szeliski, R. (2002, 01 Apr). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*(1), 7–42. Retrieved from `https://doi.org/10.1023/A:1014573219977` doi: 10.1023/A:1014573219977

Shi, J. & Tomasi, C. (1993). *Good features to track* (Tech. Rep.). Ithaca, NY, USA.

Snavely, N. (2008). *Bundler: Structure from Motion (SfM) for unordered image collections.* http://www.cs.cornell.edu/ snavely/bundler/.

Snavely, N., Seitz, S. M. & Szeliski, R. (2008, Nov). Modeling the world from internet photo collections. *Int. J. Comput. Vision*, *80*(2), 189–210. Retrieved from http://dx.doi.org/10.1007/s11263-007-0107-3 doi: 10.1007/s11263-007-0107-3

Song, S., Chandraker, M. & Guest, C. C. (2013, May). Parallel, real-time monocular visual odometry. In *2013 ieee international conference on robotics and automation* (p. 4698-4705). doi: 10.1109/ICRA.2013.6631246

Steinbruecker, F., Sturm, J. & Cremers, D. (2011). Real-time visual odometry from dense rgb-d images. In *Workshop on live dense reconstruction with moving cameras at the intl. conf. on computer vision (iccv).*

Thrun, S. (2010, Apr). Toward robotic cars. *Communications of the ACM*, *53*(4), 99–106. Retrieved from http://doi.acm.org/10.1145/1721654.1721679 doi: 10.1145/1721654.1721679

Tomasi, C. & Kanade, T. (1991). *Detection and tracking of point features* (Tech. Rep.). International Journal of Computer Vision.

Torr, P. H. S. & Zisserman, A. (2000). MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, *78*, 138–156.

Tsai, R. (1987, August). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, *3*(4), 323-344. doi: 10.1109/JRA.1987.1087109

Westoby, M., Brasington, J., Glasser, N., Hambrey, M. & Reynolds, J. (2012). 'structure-from-motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, *179*, 300 - 314. Retrieved from http://www.sciencedirect.com/science/article/pii/S0169555X12004217 doi: http://dx.doi.org/10.1016/j.geomorph.2012.08.021

Wu, C. (2011). *VisualSFM: A visual structure from motion system.* http://ccwu.me/vsfm/.

Wu, F. C., Zhang, Q. & Hu, Z. Y. (2011, 01 Jan). Efficient suboptimal solutions to the optimal triangulation. *International Journal of Computer Vision*, *91*(1), 77–106. Retrieved from https://doi.org/10.1007/s11263-010-0378-y doi: 10.1007/s11263-010-0378-y

Zeng, Y. & Klette, R. (2013). Multi-run 3d streetside reconstruction from a vehicle. In R. Wilson, E. Hancock, A. Bors & W. Smith (Eds.), *Computer analysis of images and patterns: 15th international conference, caip 2013, york, uk, august 27-29, 2013, proceedings, part i* (pp. 580–588). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-40261-6_70 doi: 10.1007/978-3-642-40261-6_70

Zhang, J. & Singh, S. (2014, July). LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and systems conference.* Berkeley, CA.

Zhang, J. & Singh, S. (2015, May). Visual-lidar odometry and mapping: Low drift, robust, and fast. In *Ieee international conference on robotics and automation.* Seattle, WA.

Zhang, Z. (2000, Nov). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330–1334.

doi: 10.1109/34.888718

Zhong, H. & Wildes., R. (2013, May). Egomotion estimation using binocular spatiotemporal oriented energy. In *British machine vision conference* (pp. 62.1–62.12).

Index

3D, 18

algorithm
- eight-point, 55
- five-point, 56, 66
- iterative-closest point, 56
- Levenberg-Marquardt, 61, 104

ALS, 19
appearance-based, 24
AR, 23, 28

BA, 22, 103
BoW, 99
BRIEF, 101
bundle adjustment, 103

constraint
- epipolar, 55

covisibility, 98

dead reckoning, 22
decomposition
- singular value, 37

DEM, 19
disparity, 42, 49
distance
- Mahalanobis, 61
- Sampson, 73

DLT, 38, 40, 51, 117
DR, 22

ego-vehicle, 19
egomotion estimation, 23, 51
epipolar
- constraint, 73
- distance, 66

error
- reprojection, 29, 41

estimation
- maximum likelihood, 41

FAST, 101
feature-based, 24
FED, 132
filtering, 64

geometry
- epipolar, 42

GPS, 23

ICP, 56, 92
IMU, 22

Kalman
- filter, 23
- gain, 82

KAZE, 132
keyframe
- matched, 102

LiDAR, 19, 92
LM, 104, 121
loop closure, 23

matrix
- covariance, 74
- essential, 55, 56, 66
- fundamental, 55, 72
- Hessian, 61, 84, 122
- Jacobian, 36, 43, 61, 78, 104, 122

motion, 33, 35

odometry
- visual, 22
- wheel, 23

ORB, 101, 130

PnP, 51, 75
Poisson surface reconstruction, 108
pose, 21
projective
    constraint, 95

RANSAC, 31, 46, 65, 78, 79
RPE, 29, 41, 74, 78, 123

SAD, 48, 50, 75
scene structure, 32
SfM, 21
SIFT, 129
SLAM, 22, 32, 64
SSD, 48, 50, 75
SURF, 130
SVD, 37, 38, 52, 55, 78

tf-idf, 101
TLS, 19
topology, 87
transform
    direct linear, 38
triality, 33
triangulation, 35
    mid-point, 40

uncertainty, 36, 71, 83

V-SLAM, 22
VO, 23, 54, 63
    real-time direct, 58
vocabulary, 97