3D localization for unmanned vehicles using visual inputs

Mou, Wei

2017

Mou, W. (2017). 3D localization for unmanned vehicles using visual inputs. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/70516>

<https://doi.org/10.32657/10356/70516>

3D localization for Unmanned Vehicles using Visual Inputs

Wei Mou

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2015

Abstract

This thesis describes our efforts to tackle the 3D localization problem for unmanned vehicles using only visual data. Given the video stream of a camera, we wish to estimate the location of the camera accurately in real-time.

We propose an indoor visual odometry system with a RGB-D camera pointing at the ceiling. The term visual odometry is chosen for its functional similarity with wheel odometry which incrementally estimates the position of a robot by counting the number of turns of its wheels over time. Similarly, visual odometry estimates the position of the robot by integrating its motion changes inferred from the images that captured by the onboard cameras. The main contribution of this algorithm is the introduction of principal direction detection that can greatly reduce error accumulation problem in most visual odometry estimation approaches. The proposed approach can be operated in real-time and it performs well even with cameras disturbance.

For robots working in outdoor environments, an efficient visual odometry system is developed. Keypoints are detected using the FAST detector. The proposed feature descriptor is designed in such a way that it is not only invariant to rotation and illumination changes but also the difference between two descriptors can be computed very efficiently using Intel Streaming SIMD Extensions (SSE) instruction. The feature matching process is accelerated using prior statistical analysis of maximum and minimum feature displacements. Experimental results show that the proposed system can perform accurate visual odometry very efficiently in outdoor environments.

In order to localize distant objects on the sea surface, an automatic self-calibration approach for wide baseline stereo cameras using sea surface images is introduced. Compared to the traditional stereo calibration method using calibration pattern, the proposed self-calibration approach automatically estimate cameras' rotation matrices for stereo rig using the sea horizon and a point at infinite distance.

Dedicated to my wife Xiaoli and my parents.

Robotics is a collaborative work, and this report would not have been possible without the help and support from many people. First and foremost, I would like to express my deepest gratitude to my supervisor Prof. Wang Han for being an excellent advisor and mentor. I am grateful not only for his enduring support and inspiration but also for his integrity, discipline, and attention to details, which have all influenced my life.

I am also very thankful to Prof. Gerald Seet. He provided me with helpful comments and critical remarks. I appreciate all the supports and encouragements he gave to me.

My labmates in ARRL deserve much thanks for making my graduate life memorable. I wish to thank Zhou Lubing, Ying Ying and Wei Zhuo for their help and inspiring suggestions when I just started my PhD study. To Yuan Shenghai and Soner Ulun with whom we had lots of talks about research, programming skills and more. To Dr. Bok-Suk Shin who has been liberal in sharing her experiences and offering advices on my research and paper writing. To Mou Xiaozheng, Sui Jiacheng and Zhao Wei, Yang Shuai for their friendship and supports during my research. Many thanks goes to Hendra Suratno with whom I accomplished many steps of my projects and he helped me a lot on experiments.

Last but certainly not least, this thesis would not have been possible without the love and support from my family. I am deeply grateful to my father Liu Xinsheng, my mother Sun Yanqin for all love, support and encouragement throughout my life. Special thanks goes to my wife Shi Xiaoli for love and massive support and great patience in my life. I dedicate this thesis to them.

Contents

List of Figures	vii
Author's Publications	1
1 Introduction	3
1.1 Six Degree of Freedom visual motion estimation	5
1.1.1 Visual odometry on ceiling images using RGB-D camera . .	6
1.2 Self-calibration of extrinsic parameters for stereo rig	9
1.3 Contributions	13
1.4 Thesis outline	14
2 Background and Literature Review	17
2.1 Camera Model	17
2.1.1 Intrinsic Parameters	18
2.1.2 Extrinsic Parameters	19
2.1.3 Projective Model	21
2.1.4 Distortion Model	22
2.2 Stereo Vision	23
2.2.1 Ideal Stereo Model	23
2.2.2 Stereo Rectification	25
2.3 Stereo Calibration with Checkerboard	28
2.3.1 Intrinsic Parameters Calibration	28
2.3.2 Extrinsic Parameters Calibration	32
2.4 Estimate extrinsic parameters from two images	33
2.4.1 Fundamental matrix estimation	35

CONTENTS

2.4.2	Extract extrinsic parameters from essential matrix	37
2.5	Homography estimation	39
2.5.1	Algorithms for homography estimation	40
2.5.2	Robust homography estimation	42
2.6	Visual odometry estimation	43
2.6.1	Monocular visual odometry	44
2.6.2	Stereo visual odometry	46
2.7	Motion estimation for stereo visual odometry	47
2.7.1	2D-to-2D motion estimation	48
2.7.2	3D-to-3D motion estimation	49
2.7.3	3D-to-2D motion estimation	50
2.7.4	Pose estimation given camera motion	51
2.7.5	FOVIS: a stereo visual odometry estimation approach . . .	53
2.8	Cost function minimization methods	54
2.8.1	Newton iteration	54
2.8.2	Gauss-Newton method	56
2.8.3	Gradient descent	57
2.8.4	Levernberg-Marquardt method	58
2.9	Conclusion	58
3	Motion Estimation on ceiling vision	61
3.1	Keypoint feature-based visual odometry	62
3.1.1	Estimation of ego motion	63
3.1.2	Experimental Results of keypoint-based of visual odometry	66
3.2	Visual odometry using principal direction	68
3.2.1	Principal direction detection	70
3.2.2	Visual Odometry Implementation	76
3.2.3	Experiments	78
3.3	Conclusion	84

4	Efficient Visual Odometry using Stereo Camera	87
4.1	Feature Detection and Descriptor Extraction	88
4.1.1	Feature detection	88
4.1.2	Descriptor extraction	89
4.2	Feature matching	90
4.3	Motion estimation	93
4.4	Experiments and discussions	94
4.5	Concluding Remarks	99
5	Self-calibration Approach for Wide Baseline Stereo Cameras using Sea Surface Images	101
5.1	Rotation parameters estimation	103
5.1.1	Derive θ (rotation around Y axis) from the horizon	107
5.1.2	Derive α (rotation around X axis) from horizon	108
5.1.3	Derive β (rotation around Z axis) from point at infinity	109
5.1.4	Rotation parameters refinement	110
5.2	Effects of translation vector	114
5.3	Experimental results	116
5.4	Conclusion	128
6	Conclusions and Future work	131
6.1	Conclusions	131
6.2	Future work	135
A	Robust Homography Estimation based on Non-linear Least Squares Optimization	137
A.1	Introduction	137
A.2	Cost Function Formulation	142
A.3	Sparsity of the system	147
A.4	Experiments and Evaluations	148
A.4.1	Synthetic noise	149
A.5	Experiments on real life images	155
A.6	Conclusion	158

CONTENTS

References	159
------------	-----

List of Figures

1.1	Mobile robots applications.	4
1.2	Kinect camera and captured images	7
1.3	Disparity changes corresponding to the same depth change with different baseline.	11
1.4	Calibration image pair using calibration board with size of 5m x 4m.	12
1.5	Feature correspondences using RANSAC.	13
2.1	The pinhole camera model.	19
2.2	The relationship between world and camera coordinate system. . .	20
2.3	Illustration of image distortion.	22
2.4	Ideal model of stereo vision.	24
2.5	Stereo rectification.	25
2.6	Stereo rectification result.	27
2.7	A sample 2D calibration board.	29
2.8	Epipolar geometry.	34
2.9	Four possible solutions for essential matrix decomposition.	38
2.10	Epipolar geometry.	49
2.11	Rigid transformation between three different frames	52
2.12	Slow convergence due to zig-zaging property of gradient descent. .	57
3.1	Kinect mounted on iRobot Create platform.	62
3.2	SURF features matching between 2 frames corresponding to 2 con- secutive view of the robot	64
3.3	Visual odometry estimation results in RRC.	67
3.4	Lines extraction of ceiling image	70

LIST OF FIGURES

3.5	Lines extraction under perspective distortion	72
3.6	System overview.	76
3.7	Robot trajectory with disturbance	79
3.8	Comparison of proposed visual odometry approach and conventional odometry.	80
3.9	Comparison of visual odometry using feature-matching based approach and principal direction based approach.	81
3.10	Gmapping results when robot moves straight forward	82
3.11	Gmapping results when robot moves along "∞" shaped loop	83
4.1	Standard sobel kernel layout.	90
4.2	Smoothing and feature detection kernel layout.	91
4.3	Designed descriptor layout	91
4.4	Circle matching of features in current and previous, left and right images.	93
4.5	The robot platform and stereo camera we used to conduct the experiments.	95
4.6	Feature correspondences of previous left and current left frames	96
4.7	Visual odometry estimation result in the real-world experiment	97
4.8	Visual odometry comparison.	98
5.1	Flowchart of proposed rotation estimation approach.	104
5.2	The camera coordinate system.	106
5.3	Relation between infinity point and β	109
5.4	Effects of translation on disparity	116
5.5	Stereo rig mounted on an unmanned surface vehicle.	117
5.6	Horizon detection results.	117
5.7	A sample calibration image pair.	118
5.8	Convergence rate of photometric error minimization process on multiple scales.	119
5.9	A sample rectified image pair with horizontal lines.	120
5.10	The average intensity for each row of down-sampled images.	121
5.11	Comparison with checkerboard-based approach.	123
5.12	Comparison with checkerboard-based approach.	124

LIST OF FIGURES

5.13	Intensity difference and stereo density of rectified sequences. . . .	125
5.14	Intensity difference and stereo density of rectified sequences. . . .	126
5.15	Intensity difference of synthetic image sequences.	127
5.16	Distance estimation to target boat.	129
A.1	Sparse structure example of Jacobian	149
A.2	Value of $\mathbf{s}_i$ in the case of 515 outliers.	151
A.3	Keypoint correspondences between original and transformed images	152
A.4	Distribution of $\mathbf{C}$ which represents the contribution of each match- ing to the optimization problem.	152
A.5	Performance comparison with RANSAC and LMedS.	153
A.6	Number of iterations needed before optimization process converges	154
A.7	Oxford VGG Graffiti Images. Green dots are detected SIFT features.	156

Author's Publications

- [1] H. Wang, W. Mou, M. Lau, G. Seet, and D. Wang, "Mobile robot ego motion estimation using ransac-based ceiling vision," in *24th Chinese Control and Decision Conference*, 2012, pp. 1939–1943.
- [2] H. Wang, W. Mou, H. Suratno, G. Seet, M. Li, M. Lau, and D. Wang, "Visual odometry using rgb-d camera on ceiling vision," in *IEEE International Conference on Robotics and Biomimetics*, 2012, pp. 710–714.
- [3] H. Wang, W. Mou, G. Seet, M. Li, M. Lau, and D. Wang, "Real-time visual odometry estimation based on principal direction detection on ceiling vision," *International Journal of Automation and Computing*, vol. 10, no. 5, pp. 397–404, 2013.
- [4] L. Zhou, H. Wang, and W. Mou, "Robust face detection and tracking under natural conditions," in *IEEE International Conference on Robotics and Biomimetics*, 2013, pp. 934–939.
- [5] W. Mou, H. Wang, G. Seet, and L. Zhou, "Robust homography estimation based on non-linear least squares optimization," in *IEEE International Conference on Robotics and Biomimetics*, 2013, pp. 372–377.
- [6] W. Mou, H. Wang, and G. Seet, "Efficient visual odometry estimation using stereo camera," in *IEEE International Conference on Control and Automation*, 2014, pp. 1399–1403.
- [7] H. Wang, X. Mou, W. Mou, S. Yuan, S. Ulun, S. Yang, and B.-S. Shin, "Vision based long range object detection and tracking for unmanned surface

AUTHOR'S PUBLICATIONS

- vehicle,” in *IEEE International Conference on Robotics, Automation and Mechatronics*, 2015, pp. 101–105.
- [8] H. Wang, W. Mou, X. Mou, S. Yuan, S. Ulun, S. Yang, and B.-S. Shin, “An automatic self-calibration approach for wide baseline stereo cameras using sea surface images,” *Unmanned Systems*, vol. 3, no. 4, pp. 277–290, 2015.
- [9] Y. Yan, B.-S. Shin, X. Mou, W. Mou, and H. Wang, “Efficient horizon detection on complex sea for sea surveillance,” *International Journal of Electrical, Electronics And Data Communication*, vol. 3, no. 12, pp. 49–52, 2015, (Best Paper Awarded).
- [10] W. Mou, S. Yang, H. Wang, and S. S. Ge, “Place recognition by combining multiple feature types with a modified vocabulary tree,” in *International Conference on Image and Vision Computing New Zealand*, 2015.
- [11] S. Yang, W. Mou, and H. Wang, “Place recognition using multiple feature types,” *Advances in Robotics and Automation*, 2015, (Accepted).

Chapter 1

Introduction

Autonomous mobile robots, self-driving car and service robots as illustrated in Figure 1.1, are no longer dreams of the future. Over the last decades, there has been a substantial progress in research, and applications of mobile robots due to rapid developments on sensor technologies, and mobile robots are expected to have a significant impact on our society and economy in the future. Compared to human beings, robots have many nice properties: They can easily be replaced, they have no fatigue problem and can be used in hazardous environments such as poisonous, high temperature or risky for humans. Equipped with lifting gears and other working tools they are often stronger and more precise than human beings. Furthermore, autonomous wheelchairs have been designed to assist people in hospital and some others are designed to help human beings in daily life, such as cooking [1] and cleaning [2].

One basic task any mobile robot needs to perform is to understand its current status such as position or orientation and so on. The common challenge for various kinds of mobile robots is that robots operating in unstructured environment need to deal with unknown surroundings and the locations of themselves. In some applications such as autonomous vacuum cleaners, the robots sometimes only alternate their directions whenever they bump into obstacles. However, in most cases, autonomous mobile robots need to do planning before the operation is applied. For planning, the robots need to maintain a certain representation of the environment as well as the current locations which is called simultaneous localization and mapping (SLAM). The requirement for SLAM problem is not only

1. INTRODUCTION

(a) Curiosity (Mars rover).

(b) Lexus RX450h retrofitted as a Google driverless car.

(c) Fujitsu service robot

Figure 1.1: Mobile robots applications.

about the accuracy but also about whether the estimated status is up-to-date. A good motion estimation is extremely valuable for many applications. Various sensors have been mounted on mobile robots to estimate its position and this position estimation process through data gathered by moving sensors is called odometry. The sensors can be wheel encoders, laser range finders, and cameras. 2D or 3D laser can generate very accurate map for surroundings, however, the high cost makes it difficult for massive commercialization. On the other hand, attention to digital cameras such as webcam, mobile phones with built in camera, and digital SLR (single-lens reflex) cameras have greatly increased due to their low cost. Barry Hendy of Kodak Australia presented Hendy's Law [3] at the Photo Marketing Association conference in 1998: The number of pixels per dollar found in digital cameras will double every year. Based on the "Hendy's Law", the resolution of the camera is becoming higher while its cost becoming lower which means more accurate measurements can be obtained from cheaper cameras. Besides cost, there are many more reasons that cause a rapidly growing interest in visual motion estimation. Visual navigation is most similar to human navigation and can provide more cues for motion estimation such as color and texture [4]. Moreover, visual inputs provide the possibilities of performing tasks similar to the way human brain perform the tasks, such as object recognition [5] or object tracking [6] and so on. These possibilities are important for cognitive tasks but are nearly impossible for other sensory data. Hence in this thesis, we

focus on getting motion estimation from purely visual inputs.

1.1 Six Degree of Freedom visual motion estimation

Visual odometry is the process of incrementally estimating the 6 Degree of Freedom (DoF) camera pose based on the analysis of captured images. The term Visual odometry is firstly introduced by Nister in 2004 in his paper [7]. The word “odometry” came from *wheel odometry*, which estimates the vehicle position changes over time by counting the number of turns of its wheels. However, the wheel odometry approach has several limitations compared to visual odometry. Firstly, the precision depends on very exact measurements of the robot wheels and wheel base. Secondly, it is unreliable in rough terrain (wheels tend to slip and sink) or the robot may be lifted and moved.

Visual odometry offers a natural complement to such limitations. It promises several advantages: It has lower drift rates and can provide a full 6-DOF motion estimation. Also it is not sensitive to ground mechanics. Last but not least, camera systems are much cheaper compared to high accuracy inertial measurement units (IMUs). Visual odometry is especially useful when GPS is not available, such as in indoor or under-water environments.

Successful approaches by combining visual odometry and GPS inputs have been demonstrated in [8] and [9]. The incremental motion estimation is fused with the GPS sensor using a Kalman filter to prevent long-term drifts. However, in GPS-denied environments such as indoor, this kind of approaches are not feasible.

Bundle adjustments are applied in many visual odometry or visual SLAM literatures ([10] [11] [12]). Bundle adjustment is the problem of refining a visual reconstruction to produce jointly optimal 3D structure and camera pose estimates. The parameter estimates are found by minimizing some cost function using iterative method that quantifies the model fitting error, and jointly that the solution is simultaneously optimal with respect to both structure and camera motions. Although bundle adjustment can be used to refine the local estimate of

1. INTRODUCTION

the trajectory, the computation time grows with the number of images in local windows. Visual odometry focuses on estimating the 3-D motion of the camera sequentially as a new frame arrives in real time, and it is often used as building blocks of visual SLAM or bundle adjustment. As a result, real-time performance is crucial for visual odometry estimation. The standard visual odometry estimation pipeline involves the process of feature detection and matching which consumes much computation time. Speeding up this process will improve the real-time performance for any system involving visual odometry. In chapter 4, our efforts to maximize the efficiency of this process on a modern CPU is described.

1.1.1 Visual odometry on ceiling images using RGB-D camera

The advantage of stereo vision over other range measuring devices is that it acquires high resolution color images of the scene together with the depth information. The main idea is to find corresponding points in images from both cameras and calculate the distance using triangulation. The major problem is to identify the correspondences precisely which could be difficult for image parts without many visual details and it also has high computational demands.

The Time-of-Flight (ToF) camera is another choice to acquire depth of the scene. ToF cameras work by measuring the phase-delay of reflected infrared (IR) light. ToF cameras can be used to estimate 3D structure directly without the help of traditional computer vision algorithms. However, the ToF camera has a few drawbacks such as a blurred depth image on fast moving objects. Moreover, the precision of the measured depth and resolution of the depth image are low and its high cost makes it only suitable for lab use.

The introduction of the Microsoft Kinect camera [13] (Figure 1.2(a)), accompanied by massive sales of this product, has made widely available depth data streams, at video rate, of generic static or dynamic scenes. Although the Kinect was introduced as a gaming controller for Xbox 360 console from Microsoft, a way to use it with personal the computer was quickly discovered by the opensource community. Kinect senses the depth of the scene using structured inferred light. It consists of 3 cameras : one inferred light projector, one inferred camera and one

1.1 Six Degree of Freedom visual motion estimation

color camera and these cameras can provide the color image as shown in Figure 1.2(b), the depth map and the IR image 1.2(c) with a resolution of 640×480 . The depth sensing range is more accurate when object is within a range between 0.8 meters to 4.0 meters. The spatial resolution is 3 mm at 2 meters distance and the depth resolution is 10 mm at 2 meters distance. The FOV of Kinect is 57×43 degrees and it can overcome the disadvantages of ToF camera.

(a) Xbox360 Kinect camera

(b) Color image

(c) Infrared image

Figure 1.2: Kinect camera and captured images

Many visual odometry methods based on a RGB-D camera have been examined in the literature, and they can be classified into two categories: sparse and dense methods.

In the category of “sparse method”, some authors ([14]) use feature correspondences between two consecutive frames together with depth information acquired from RGB-D camera to estimate motion. However, the feature (such as SIFT or

1. INTRODUCTION

SURF) extraction and matching (such as [15][16]) are time consuming which is infeasible for low-latency systems. Although some patch-based approaches such as the KLT tracker [17] or PTAM [18] tracker are fast, for sparse methods, much relevant information about the scene is discarded in order to gain computational efficiency, and thus, do not exploit the sensor data optimally.

In the category of “dense method”, the authors in [19] proposed a visual odometry and 3D surface reconstruction algorithm (KinectFusion). It was one of the first systems to produce a volumetric reconstruction of a scene in real-time, with an unprecedented level of accuracy, and made dense approaches popular for RGB-D camera. Iterative closest point (ICP) approach is used to register captured point clouds in a reconstructed environment and thus estimate the camera motion. ICP-based approach is more robust compared to the feature-based approach, when the camera motion is small. However, in the case of large camera displacement, or lack of 3D depth features in the scene, the ICP-based approach is prone to failure. In [20] authors proposed an improved version of KinectFusion by incorporating a feature-based visual odometry approach to deal with the limitations of KinectFusion. The aforementioned dense approach can run in real-time only if GPUs are available for high performance computing. However, GPUs are not applicable in many cases such as for RGB-D camera mounted on small UAVs. The authors in [21], [22] and [23], proposed a dense visual odometry estimation approach from RGB-D frames directly, and the approaches can run in real-time with a single CPU core. The photo-consistency assumption is required and by minimizing the photometric error, two consecutive RGB-D frames can be aligned and thereby obtain an estimation of camera’s motion. unlike sparse methods, all information of two consecutive frames are utilized and better accuracy can be obtained.

The camera motions estimated by visual odometry are between two consecutive frames. No matter how accurate this relative motion estimation could be, the estimated motion still suffers from the drift problem, which means small errors of relative motion will accumulated overtime and result in false estimation of current camera’s pose. Various visual SLAM approaches have been proposed in the literature to correct the accumulated error (to list a few: [24][25][26][27]). How-

ever, they introduce new problems such as key-frame selection [28], loop closure detection [29], and extra computation power is required.

In an indoor environment, there is often a dominant direction (such as the features orientation on the ceiling) for each image and thus the global orientation of the camera can be determined. In order to reduce the accumulated error as well as saving computation power, instead of doing expensive visual slam or bundle adjustment, we exploit the features in the scene so that some global information of the camera's orientation can be retrieved and the accumulated errors will be confined.

1.2 Self-calibration of extrinsic parameters for stereo rig

Precise motion estimation requires precise calibration of cameras. Camera calibration is the first step for many visual localization applications in 3D computer vision in order to extract metric information from 2D images. If the calibration is not accurate, it will ruin the following steps, no matter how accurate they are. The errors will be mixed with the following errors or even be amplified. Camera calibration consists of two parts. The first part is camera modelling which models physical and optical behavior of the camera mathematically. The second part is to estimate the values of parameters in the built camera model. There are two kinds of parameters in the model which have to be considered. Intrinsic parameters describe the internal geometry and optical characteristics of the image sensor. Basically, intrinsic parameters determine how light is projected through the lens onto the image plane of the sensor. The extrinsic parameters measure the position and orientation of the camera with respect to a world coordinate system which, in turn, provides metric information with respect to the camera coordinate system.

For a monocular camera, only its intrinsic parameters need to be estimated from calibration. Various efficient approaches for camera's intrinsic calibration have been examined in the literature using either 3D or 2D calibration objects

1. INTRODUCTION

such as ([30] and [31]). Although calibration objects are different, most approaches estimate intrinsic parameters by minimizing the reprojection error between a set of known 3D points to their 2D projections on images.

Compared to a monocular camera, the calibration process for stereo cameras is more complicated because more parameters, such as the extrinsic parameters between two cameras, need to be estimated. However, knowing the baseline, stereo cameras can estimate scene depth with absolute scale within one frame, which is essential for motion estimation. Hence, stereo camera has natured advantages against monocular camera in solving motion estimation problem. For extrinsic parameters estimation, the general approach is similar to intrinsic parameters estimation: Minimize reprojection error of the know 3D points, even same calibration objects can be used [32]. Although these approaches are very convenient and popular for stereo calibration, they are only suitable for indoor and near-range applications, because during the calibration process, the reprojection error is minimized using a small calibration pattern and for near-range objects. Bigger calibration patterns or far-range objects can be used for calibration purposes [33]. The stereo rigs also suffer from the problem of unstable configuration due to poor environment, such as moisture variation or mechanical vibration and so on. In many cases, such as for a stereo rig mounted on a boat in the sea, it is almost impossible to have any stable calibration object. This requires that the extrinsic parameters for stereo rigs should be easy and efficient to calibrate and this kind of self-calibration property is an important requirement for stereo camera in automatic robotics field, especially when long-term operation is desired.

Our current research focus on unmanned surface vehicles (USV), and we are in need of a system that can estimate targets' motion and localize them on the sea surface within a certain range. In order to have proper depth estimation of distant targets (about 500 meters), a stereo rig with very wide baseline (3 meters) is used solely to accomplish the task. The reason for using wide baseline is that compared to narrow baseline, the range of disparity is larger, thus the depth estimation is more accurate. As depicted in Figure 1.3, two world points P and Q are projected onto two stereo systems with narrow and wide baseline. The depth change of P and Q are the same in both systems, however, the disparity

1.2 Self-calibration of extrinsic parameters for stereo rig

change d' in wide baseline case is bigger than d in narrow baseline case. Thus, the resolution of the disparity map in wide baseline case is higher which means the depth estimation will be more accurate.

Figure 1.3: Disparity changes corresponding to the same depth change with different baseline.

To calibrate such a wide baseline stereo system, a large calibration pattern is projected on a large screen ($5m \times 4m$ as shown in Figure 1.4). In order to capture multiple images from different orientations, at least four people or even a crane is needed. Moreover, in order to estimate the extrinsic parameters, the whole calibration pattern has to appear in both stereo image pairs. However, even with such a big checkerboard, the calibration grid can only cover a small portion of images which will result in inaccurate estimation of extrinsic parameters. A even much bigger checkerboard can be built to calibrate such a very wide baseline stereo rig. However, that would require extra personnel or big equipment like crane to operate the calibration process which makes it infeasible in real applications.

Self-calibration is the process of automatic estimation of extrinsic camera parameters of a stereo rig using almost arbitrary image pairs. The advantages of self-calibration is that, no special calibration pattern is required and the calibration can be done even when the cameras are in use. Image pairs are taken by

1. INTRODUCTION

Figure 1.4: Calibration image pair using calibration board with size of 5m x 4m.

the stereo cameras with known intrinsic parameters. In [34], [35] and [36], point features are detected from both images and their correspondences are established. Given the feature correspondences, the fundamental or essential matrix can be obtained and can be used to derive rotation and translation between two cameras up to a certain scale. In [37], the authors first estimated the essential matrix in order to find camera motion between two frames. After that, a non-linear refinement is applied by incorporating depth ordering constraints. The experiments using a static camera were carried out in a small indoor environment. In [38], authors assume a proper initial guess of the extrinsic parameters is available (up to a few degrees for camera orientation) and an adapted iterated extended Kalman filter is applied to refine the calibration recursively and continuously. There are also other methods that use line correspondences [39], partial pose measurements [40] and line-to-plane correspondence [41]. Some authors prefer to use the landmarks of the road [42], such as traffic lines ([43]), to make the calibration process easier and update the extrinsic parameters continuously. In [44], the authors calibrate extrinsic parameters by using the changes of the road profile in front of the stereo rig.

Although these approaches are automatic and flexible, some of them require finding proper feature correspondences, which makes them infeasible in our application. In the case of a stereo rig with very wide baseline, the same 3D world point

Figure 1.5: Feature correspondences using RANSAC.

may have very different appearance in left and right images since it is viewed in both cameras from very different perspectives. Hence, to find the correct feature correspondences automatically is already a very challenging task. Compared to other scenarios, finding feature correspondences automatically is even more challenging when using sea images. As shown in Figure 1.5, ocean surface is not a feature-rich environment and prone to mismatch due to appearance similarity. As a result, even though feature correspondences can be obtained through RANSAC [45], most correspondences will be clustered on the boat which will result in unreliable results. The sea surface is not as static as road surface, so it is inappropriate to do calibration by using landmarks or by observing changes of sea surface. To deal with such problem on the sea surface, an automatic self-calibration approach for stereo camera which requires no special-designed calibration objects is discussed in chapter 5.

1.3 Contributions

The main contributions of the thesis are summarized below:

- For an indoor environment, a principal direction detection approach using Kinect camera is proposed that can greatly reduce error accumulation in most visual odometry estimation approaches. The proposed approach can be operated in real-time and it performs well even with unpredictable disturbance in camera motion.

1. INTRODUCTION

- A specially designed feature descriptor is proposed which can improve efficiency of stereo visual odometry system. The feature descriptor is designed in such a way that it is not only invariant to rotation and illumination changes but also the difference between two descriptors can be computed very efficiently using Intel Streaming SIMD Extensions (SSE) instruction. The feature matching process is accelerated using prior statistical analysis of maximum and minimum feature displacements.
- To localize the targets on the sea surface, an automatic self-calibration approach for wide baseline (3 meters) stereo cameras using sea surface images is introduced. Instead of using specially designed calibration pattern or establishing feature correspondences, the proposed self-calibration approach can automatically estimate extrinsic parameters for stereo rig with wide baseline, using the sea horizon and a point at infinite distance.
- A cost function for robust 2D projective transformation or homography estimation approach is proposed. In order to estimate homography, feature correspondences need to be firstly established using descriptor similarity. False correspondences are inevitable and need to be identified before estimating the homography. The proposed cost function is parameterized in such a way that incorrect correspondences can be simultaneously identified while the homography is computed. This allows us to skip the RANSAC-based outlier rejection step and speed up motion estimation process.

1.4 Thesis outline

Chapter 2 details the camera model and the mathematics of projective geometry that will be used in the thesis. The background of camera calibration, and 2D or 3D motion estimation using visual inputs, are also introduced. This chapter provides some of the fundamental equations upon which our visual odometry systems, self-calibration of the stereo camera, and homography estimation approach are built.

Chapter 3 introduces a visual odometry estimation approach in an indoor environment using ceiling vision. The main contribution in this algorithm is the

introduction of principal direction detection (that can greatly reduce error accumulation problem present in most visual odometry estimation approaches). The principal direction is defined based on the fact that our ceiling is filled with artificial vertical and horizontal lines and these lines can be used as a reference for the current robot's heading direction. The proposed approach can be operated in realtime and it performs well even with camera's disturbance. A moving (low-cost) RGB-D camera (Kinect), mounted on a robot, is used to continuously acquire point clouds. Iterative Closest Point (ICP) is the common way to estimate current camera position by calculating the translation and rotation to the previous frame. However, its performance suffers from data association problem or it requires pre-alignment information. Unlike ICP, the performance of the proposed approach does not rely on data association knowledge. Using this method, two point clouds are pre-aligned. Hence, we can use ICP to fine-tune the transformation parameters and to minimize registration error. Experimental results demonstrate the performance and stability of the proposed system under camera's disturbance in real-time.

Chapter 4 introduces an efficient visual odometry estimation approach using stereo camera. In this chapter a visual odometry system that estimate the motion of a stereo camera from consecutive stereo image pairs is presented. No other sensors or prior knowledge of the scene is needed. The point features are extracted and matched between image pairs. Given feature correspondences, the camera motion is estimated by minimizing the reprojection error of successive images. A customized feature descriptor that can make feature matching process extremely fast while preserving the matching reliability is proposed in this chapter. Hence, we can include as many features as possible into the optimization process which leads to improvement of motion estimation accuracy. Real-world experiments are done to demonstrate the ability and efficiency of the system.

Chapter 5 introduces an automatic self-calibration approach for wide baseline stereo cameras using sea surface images. In contrast to the traditional stereo calibration method using calibration pattern, we propose a self-calibration approach that can easily estimate cameras' rotation matrices for stereo rig with very wide baseline (3 meters) using sea horizon and a point at infinite distance. Compared with conventional techniques which require expensive equipment or

1. INTRODUCTION

manpower, our approach only needs a pair of sea images. Moreover, special calibration patterns and feature correspondences are not required. As a result, it is flexible and very easy to implement. Real world experiments demonstrate the feasibility of our approach.

Chapter 6 concludes the thesis and looks into the possible future work. The contributions of the thesis are presented in 4 aspects: a self-calibration of extrinsic parameters, a cost function for simultaneous homography estimation and outlier detection, improved efficiency for stereo visual odometry, and principal direction detection (to deal with visual odometry drift).

Appendix A describes our approach for simultaneous Homography estimation and outlier detection. The homography between image pairs are normally estimated by minimizing a suitable cost function given 2D keypoints correspondences. The correspondences are typically established using descriptor distance of keypoints. However, the correspondences are often incorrect due to ambiguous descriptors which can introduce errors into following homography computing step. There have been numerous attempts to filter out these erroneous correspondences, but, it is unlikely to always achieve perfect matching. To deal with this problem, we propose a non-linear least squares optimization approach to compute homography such that false matches have no or little effect on computed homography. Unlike normal homography computation algorithms, our method formulates not only the keypoints' geometric relationship but also their descriptor similarity into cost function. Moreover, the cost function is parameterized in such a way that incorrect correspondences can be simultaneously identified while the homography is computed. Experiments show that the proposed approach can perform well even with the presence of a large number of outliers.

Chapter 2

Background and Literature Review

Stereo vision is critical for many applications from stereo visual odometry to 3D scene reconstruction. One of the most prominent advantages of stereo vision is that it can provide not only the color images of two cameras but also the 3D structures of the scene which are important for many object tracking or detection tasks. Stereo camera calibration is necessary to match image pixels captured by stereo cameras to world 3D coordinates. Stereo camera calibration is the process that estimates camera parameters using either direct or iterative methods. With calibrated stereo cameras, scene depth can be calculated through simple triangulation which makes 6-DOF motion estimation possible.

2.1 Camera Model

This section introduces the notation that we use to describe the camera projection and transformation that is going to be used throughout the thesis. Camera modelling is to approximate a physical sensing device using mathematical formulation. Many camera models has been proposed, depending on the desired accuracy of the applications [50]. The pinhole camera model [51] is used in this thesis. It defines a linear projection from 3D world points to $2D$ image pixels without modelling the lens distortion. In general, the parameters to define a camera can be categorized into two groups: the first group consists of intrinsic

2. BACKGROUND AND LITERATURE REVIEW

parameters which together describe the geometric property of a camera, while, the other group consists of extrinsic parameters which represent the position and orientation of the camera in world coordinate system.

2.1.1 Intrinsic Parameters

The camera intrinsic parameters describe how to project a 3D world scene onto the 2D image plane. Figure 2.1 shows a pinhole camera model with center of projection O and the principal axis Z . The image plane is the place that the 3D world is projected on. The intrinsic parameters can be put in one intrinsic matrix K that transforms 3D camera coordinates to 2D homogeneous image coordinates as:

$$K = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

A 3D world point $P \in \mathbb{R}^3$ with coordinate (X, Y, Z) is imaged on the cameras image plane at a 2D image point $p \in \mathbb{R}^2$ with coordinate (u, v) .

The focal length affects the object size in the image and it is measured in pixels. In practice, f_x and f_y may differ for reasons like distortion or flaws in the digital camera sensor which will result in non-square pixels. However, in the pinhole camera model that we used, f_x and f_y have same value f which is the distance from O to image plane as illustrated in Figure 2.1.

(u_0, v_0) is the principal point of the camera which is the intersection of principal axis Z and image plane as depicted in Figure 2.1. And same as focal length, the principal point is also measured in pixels. For most images, the principal point is almost located at the center of image. However, for a camera with shorter focal length or cropped images, the principal point can have significant shifts away from the image center.

Axis Skew s results in shear distortion in the projected image. For digital cameras, the skew is normally zero.

2.1.2 Extrinsic Parameters

The extrinsic parameters define the position and orientation of the camera in the world coordinate system. Several coordinate systems are specified in order to understand the effects of extrinsic parameters:

- **World coordinate system** is a 3D coordinate system for the real world and it is usually used as reference frame for camera position.
- **Camera coordinate system** is a 3D coordinate system that is used to measure positions of objects in the scene with respect to the camera.
- **Image coordinate system** is a 2D coordinate system in the image plane. It is used to measure the position of the projection of a 3D object in the image.

Figure 2.1: The pinhole camera model.

2. BACKGROUND AND LITERATURE REVIEW

As shown in Figure 2.2, the camera is placed in world coordinate system. The camera coordinate system are not aligned with world coordinate system. This transformation between world and camera coordinate system can be expressed using a *rotation matrix* $\mathbf{R} \in SO(3)$ and a *translation vector* $\mathbf{t} \in \mathbb{R}^3$ as:

$$\bar{O}_c = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix} \bar{O}_w \quad (2.2)$$

Figure 2.2: The relationship between world and camera coordinate system.

A 3D rotation matrix $\mathbf{R}$ can be decomposed into three 2D rotation matrix $\mathbf{R}_x(\alpha)$, $\mathbf{R}_y(\beta)$ and $\mathbf{R}_z(\theta)$ that describe the rotations around x , y and z axes accordingly:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (2.3)$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (2.4)$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

where α , β and θ are the rotation angles around x , y and z axes accordingly. Thus, $R = R_z(\theta)R_y(\beta)R_x(\alpha)$. The rotation matrix R is orthogonal matrix which means its transpose is equivalent to its inverse: $RR^T = R^TR = RR^{-1} = I$, where I represents identity matrix.

The translation vector $\mathbf{t}$ represents the shift of origin from one coordinate system to another coordinate system. Thus, to shift from world coordinate system to camera coordinate system, the translation vector is: $\mathbf{t} = origin_{world} - origin_{camera}$.

The transformation of the scene point is equivalent to the inverse transformation of the camera. For example, we can rotate the camera coordinate system by an angle θ which is equivalent to counter-rotating the scene point around the origin of that coordinate system by the same angle θ . Let $P_w(x_w, y_w, z_w)$ be a point in the world coordinate system and $P_c(x_c, y_c, z_c)$ represents the same 3D point in the camera coordinate system. Then we can transform the point in world coordinate system to the camera coordinate system as:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} R^T & -R^T \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \quad (2.6)$$

2.1.3 Projective Model

Given the intrinsic and extrinsic parameters, the projection of a 3D world point $X \in \mathbb{R}^3$ onto 2D image plane of a camera is:

$$sU = PX \quad (2.7)$$

where the image point $U \in \mathbb{R}^2$ in the projective coordinate is $U = [u, v, 1]^T$. s is a scale factor. P is a 3×4 camera projection matrix formed by combining intrinsic and extrinsic parameters as:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KR[\mathbf{I}|\mathbf{t}] = K \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.8)$$

where $\mathbf{r}_i$ represents the i th column of rotation matrix R .

2. BACKGROUND AND LITERATURE REVIEW

Figure 2.3: Illustration of image distortion.

2.1.4 Distortion Model

The projection process in a pinhole camera model is simple and linear as defined in 2.8, because we assume there is no distortion in the image. However, for real cameras, this assumption is not valid, especially for cheap lenses. Effects of distortion are illustrated in Figure 2.3. In Figure 2.3(b), **distorted** and **ideal** are the projected point with and without distortions.

Both radial and tangential distortions can be represented by a distortion coefficient vector $k_c \in \mathbb{R}^5$. (x_d, y_d) and (x_u, y_u) are the distorted and undistorted image points respectively and they can be related through distortion coefficient as:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + k_{c1}r^2 + k_{c2}r^4 + k_{c5}r^6) \begin{bmatrix} x_u \\ y_u \end{bmatrix} + \begin{bmatrix} 2k_{c3}x_u y_u + k_{c4}(r^2 + 2x_u^2) \\ 2k_{c4}x_u y_u + k_{c3}(r^2 + 2y_u^2) \end{bmatrix} \quad (2.9)$$

where $r = x_u^2 + y_u^2$. The effect of radial distortion is usually much larger than the distortion effect caused by tangential distortion, and the further the distance to the principal point the larger the radial distortion effect will be.

2.2 Stereo Vision

2.2.1 Ideal Stereo Model

Usually, stereo vision systems have two cameras separated in space. Both cameras capture images simultaneously. It is similar to human eyes, in the way that two images captured at the same time from two different perspectives. The 3D world points are projected into both left and right images. The closer the object is to the camera, the bigger the difference in position in the images taken by the left and right cameras. The measurement of this difference is called *disparity*. Using the principle of triangulation, the depth of the scene can be recovered from the disparities.

Figure 2.4 shows the ideal model of stereo vision. The left and right cameras are aligned which means their optical axes are parallel. The Euclidean distance between left and right cameras is the baseline with a length of b . Two cameras have the same focal length f . The cameras are modeled as a perfect pinhole camera which means there is no distortion or skew in both cameras and the principal points of two cameras are at the center of each image. A world 3D point P is projected into the left and right images. Point P whose coordinate is represented as (x, y, z) and the axes of this coordinate system is aligned with left camera and its center is also located at the center of the left camera. P_l and P_r are the projection of 3D point P in the left and right images respectively. The image coordinates of P_l and P_r are represented as (u_l, v_l) and (u_r, v_r) . Thus for stereo vision, this simple model has some desirable properties. Firstly, the projection of the same 3D world point appears at the same row in both left and right images. For stereo matching, the search space is restricted to a one-dimensional line which is a horizontal line across the images captured by both cameras. Secondly, the column of the projection in the right image must be at the same column or at the left of the projection in the left image. In left camera:

$$u_l = f \frac{x}{z} \quad \text{and} \quad v_l = f \frac{y}{z} \quad (2.10)$$

In the right camera:

$$u_r = f \frac{x+b}{z} \quad \text{and} \quad v_r = f \frac{y}{z} \quad (2.11)$$

2. BACKGROUND AND LITERATURE REVIEW

Using 2.10 and 2.11 we can derive disparity d as:

$$d = u_r - u_l = f \frac{x+b}{z} - f \frac{x}{z} = f \frac{b}{z} \quad (2.12)$$

The depth of the scene is inversely proportional to stereo disparity as:

$$z = f \frac{b}{d} \quad (2.13)$$

Given scene depth z , x and y of a scene point can be computed using simple camera projection:

$$x = u_l \frac{z}{f} \quad (2.14)$$

$$y = v_l \frac{z}{f} \quad (2.15)$$

Figure 2.4: Ideal model of stereo vision.

2.2.2 Stereo Rectification

In this subsection, all images are assumed to have been undistorted using distortion coefficients. In real stereo systems, the left and right cameras are normally not coplanar. By knowing the intrinsic and extrinsic parameters of the two cameras we can apply a transformation to project the image pixels from each image plane of the camera onto a common image plane as shown in Figure 2.5. O_l and O_r are the projection centers for left and right cameras respectively. They both observing the same world point P . The projections of the 3D point P on left and right image planes are represented as P_l and P_r respectively. The projection of each camera center onto the other image plane are the *epipoles*, which are represented as E_l and E_r . The two camera centers, together with world point P form a plane called *epipolar plane*. The intersection of this plane with each image plane is *epipolar line*. A nice property of the epipolar line is that: given P_l on the left image plane, its correspondence on the right image plane will be on the epipolar line. The resulting images correspond to the ideal stereo model we

Figure 2.5: Stereo rectification.

2. BACKGROUND AND LITERATURE REVIEW

introduced in section 2.2.1 and this process is called rectification. This process can be implemented given intrinsic matrices K_l and K_r and extrinsic matrices R and $\mathbf{t}$ [52]. Firstly, for each camera, the intrinsic and extrinsic parameters can be combined in one camera matrix as:

$$M_l = K_l [\mathbf{I} \quad \mathbf{0}] \quad \text{and} \quad M_r = K_r [R \quad \mathbf{t}] \quad (2.16)$$

Secondly, if we use left camera coordinate as world coordinate, the epipoles E_l and E_r can be computed as:

$$E_l = M_l \begin{bmatrix} O_r \\ 1 \end{bmatrix} = M_l \begin{bmatrix} -R^T \mathbf{t} \\ 1 \end{bmatrix} = K_l [\mathbf{I} \quad \mathbf{0}] \begin{bmatrix} -R^T \mathbf{t} \\ 1 \end{bmatrix} = -K_l R^T \mathbf{t} \quad (2.17)$$

$$E_r = M_r \begin{bmatrix} O_l \\ 1 \end{bmatrix} = M_r \begin{bmatrix} 0 \\ 1 \end{bmatrix} = K_r [R \quad \mathbf{t}] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = K_r \mathbf{t} \quad (2.18)$$

Thirdly, the rotation matrix R is divided into two rotation matrices R_l and R_r for the left and right cameras respectively. Each camera rotate half of R so that the cameras are parallel to each other and both cameras pointing along the vector sum of where their original optical rays had been pointing.

Fourthly, we need to apply a rotation matrix R_{rec} on both cameras such that they are perpendicular to the baseline. To construct R_{rec} we start with the direction of relative translation:

$$e_1 = \frac{\mathbf{t}}{\|\mathbf{t}\|} \quad (2.19)$$

The next direction e_2 is chosen in such a way that it is perpendicular to e_1 and optical rays of the cameras using cross product:

$$e_2 = \frac{[t_y \quad t_x \quad 0]^T}{\sqrt{t_x^2 + t_y^2}} \quad (2.20)$$

The third direction vector is obtained as cross product of e_1 and e_2 :

$$e_3 = e_1 \times e_2 \quad (2.21)$$

R_{rec} that rotate two cameras such that they are perpendicular to baseline is:

$$R_{rec} = \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix} \quad (2.22)$$

Finally, both left and right images are scaled to the same focal lengths [53].

Figure 2.6 shows an rectification example of a image pair captured by a stereo rig. After rectification, the disparity of a pixel can be determined by comparing the similarity along 1D horizontal line of the image. The 3D scene structure can be estimated as discussed in section 2.2.1. In order to have such rectified images, camera calibration is necessary to get intrinsic and extrinsic parameters.

(a) Before rectification

(b) After rectification

Figure 2.6: Stereo rectification result.

The ideal stereo vision is basically achieved by means of approaches from two categories: The approaches in first category use camera calibration. The stereo

2. BACKGROUND AND LITERATURE REVIEW

cameras parameters can be determined using images with specially designed calibration pattern in them captured by two cameras from different views. Intrinsic calibration is concerned only with obtaining the physical and optical parameters of the camera ([54] and [55]). Extrinsic calibration computes the position and orientation of the camera in the scene ([39] and [56]).

However, camera calibration using specially designed pattern cannot be used in many cases due to its lack of flexibility. The approaches in the second category are based on establishing geometric relations between two images from left and right cameras respectively which is called epipolar geometry. A lot of work has been done on epipolar geometry and how it is included in the fundamental and essential matrix [57] as well as the estimation of the camera position [58]. In [59], authors analyzed the stability of the fundamental matrix due to uncertainty in the epipole computation, noise in the image point localization and so on. Robust methods to improve the computation accuracy of the fundamental matrix are proposed ([60] and [61]). The approaches from these two categories are presented in the following two sections.

2.3 Stereo Calibration with Checkerboard

Whenever working with stereoscopy, it is necessary to obtain the required intrinsic and extrinsic parameters. The intrinsic parameters do not depend on the scene viewed by the camera. The extrinsic parameters are used to represent the coordinate transformation from the world coordinate system to the camera coordinate system. These parameters can be determined through camera calibration. Most stereo calibration tools require a checkerboard to perform the calibration process. A single camera can be defined using 10 parameters (i.e. 5 intrinsic parameters and 5 radial distortion coefficients). Thus, a stereo camera model requires a total of 26 parameters (i.e. 6 extrinsic parameters and 2×10 intrinsic parameters).

2.3.1 Intrinsic Parameters Calibration

The intrinsic calibration matrix K (2.1) provides the transformation from 3D camera coordinates to 2D pixel coordinates. The calibration of intrinsic param-

2.3 Stereo Calibration with Checkerboard

eters of a camera can be done using a specially designed calibration object. In practice, a regular pattern such as a checkerboard is most commonly used as calibration pattern. In [30] calibration is done efficiently using calibration object containing several orthogonal planes. In [62], authors used a plane with an accurately known translation to the camera, to do calibration. The disadvantages of such kinds of calibration methods is that an expensive calibration object or carefully designed configuration is required. A new generation of calibration methods uses several images of a $2D$ calibration grid (Figure 2.7) that can be printed on a paper. The corner of each square is used as a calibration point and the distance between each corner is precisely measured and used in the calibration process. In [63], instead of using one view of a carefully designed 3D calibration object, author proposed a flexible calibration approach using multiple views of a flat $2D$ calibration patterns. The $2D$ calibration pattern is composed of white and black squares. This flexible approach firstly calculates a direct analytical solution of

Figure 2.7: A sample $2D$ calibration board.

2. BACKGROUND AND LITERATURE REVIEW

the parameters by assuming no lens distortion. M images of the checkerboard consisting of N corners are taken from different views. For each corner of the checkerboard images, one equation as introduced in equation 2.8 is obtained. We can safely assume that the checkerboard lies on the plane with zero depth without losing generality. As a result, equation 2.8 is simplified as:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = K \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \quad (2.23)$$

The homography that maps corners on checkerboard to the destination image plane is defined by:

$$H = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = \frac{1}{s} K [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] \quad (2.24)$$

The homography H can be computed using a set of point correspondences between source image plane and target image plane without knowing intrinsic parameters. From equation 2.24 we can obtain:

$$\mathbf{r}_1 = sK^{-1}\mathbf{h}_1 \quad (2.25)$$

$$\mathbf{r}_2 = sK^{-1}\mathbf{h}_2 \quad (2.26)$$

$$\mathbf{t} = sK^{-1}\mathbf{h}_3 \quad (2.27)$$

Because the rotation matrix is orthogonal, we have:

$$\mathbf{r}_1 \mathbf{r}_2^T = 0 \quad (2.28)$$

By substituting $\mathbf{r}_1$ and $\mathbf{r}_2$ using equation 2.25 and equation 2.26 we have:

$$\mathbf{h}_1^T (K^{-1})^T K^{-1} \mathbf{h}_2 = 0 \quad (2.29)$$

With the orthogonal property of R , we can also have:

$$\mathbf{r}_1^T \mathbf{r}_1 = \mathbf{r}_2^T \mathbf{r}_2 \quad (2.30)$$

Substitute $\mathbf{r}_1$ and $\mathbf{r}_2$ we have:

$$\mathbf{h}_1^T (K^{-1})^T K^{-1} \mathbf{h}_1 = \mathbf{h}_2^T (K^{-1})^T K^{-1} \mathbf{h}_2 \quad (2.31)$$

2.3 Stereo Calibration with Checkerboard

$(K^{-1})^T K^{-1}$ can be calculated as:

$$(K^{-1})^T K^{-1} = \begin{bmatrix} \frac{1}{f_u^2} & 0 & -\frac{c_u}{f_u^2} \\ 0 & \frac{1}{f_v^2} & -\frac{c_v}{f_v^2} \\ -\frac{c_u}{f_u^2} & -\frac{c_v}{f_v^2} & 1 + \frac{c_u}{f_u^2} + \frac{c_v}{f_v^2} \end{bmatrix} \quad (2.32)$$

For simplicity of presentation we can write:

$$(K^{-1})^T K^{-1} = A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad (2.33)$$

With equation 2.33, equation 2.29 and 2.31 have the form $\mathbf{h}_i^T A \mathbf{h}_j$ in them. As can be seen in equation 2.32, A is symmetric. Hence, $\mathbf{h}_i^T A \mathbf{h}_j$ can be written in the form of vector dot product:

$$\mathbf{h}_i^T A \mathbf{h}_j = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix}^T \begin{bmatrix} A_{11} \\ A_{12} \\ A_{22} \\ A_{13} \\ A_{23} \\ A_{33} \end{bmatrix} = \lambda_{ij}^T \mathbf{a} \quad (2.34)$$

with the definition of λ_{ij}^T , equation 2.29 and 2.31 can be combined as:

$$\begin{bmatrix} \lambda_{12}^T \\ (\lambda_{11} - \lambda_{22})^T \end{bmatrix} \mathbf{a} = 0 \quad (2.35)$$

M such equations can be stacked given M captured checkerboard images:

$$\Lambda \mathbf{a} = 0 \quad (2.36)$$

where Λ is a matrix with dimension $2M \times 6$ and $\mathbf{a}$ can be solved from the equation.

The intrinsic parameters can be calculated from $\mathbf{a}$ as:

$$f_u = \sqrt{\frac{\gamma}{A_{11}}} \quad (2.37)$$

$$f_v = \sqrt{\frac{\gamma A_{11}}{A_{11}A_{22} - A_{12}^2}} \quad (2.38)$$

$$c_u = -\frac{A_{13}f_u^2}{\gamma} \quad (2.39)$$

$$c_v = \frac{A_{12}A_{13} - A_{11}A_{23}}{A_{11}A_{22} - A_{12}^2} \quad (2.40)$$

2. BACKGROUND AND LITERATURE REVIEW

where

$$\gamma = A_{33} - \frac{A_{13}^2 + c_v(A_{12}A_{13} - A_{11}A_{23})}{A_{11}} \quad (2.41)$$

The aforementioned intrinsic parameters estimation assumed no radial distortion. Next, the radial distortion parameters are estimated directly by assuming that all other parameters are constant. Finally, a non-linear optimisation step is performed using the full model and all parameters.

2.3.2 Extrinsic Parameters Calibration

The calibration of extrinsic parameters is to find the rotation matrix R and translation vector $\mathbf{t}$ between two cameras. Intrinsic parameters of a camera are constant unless its focus is changed. To calibrate the extrinsic parameters of a stereo rig, the intrinsic parameters for both left and right cameras are calibrated individually. Multiple images of a calibration board are taken by the stereo cameras from different views. The extrinsic parameters are estimated for each of the positions of the cameras in relation to the calibration board.

For each corner of calibration grid in 3D world Q_w , in left and right camera coordinate systems, it is represented as:

$$Q_l = R_l Q_w + t_l \quad (2.42)$$

$$Q_r = R_r Q_w + t_r \quad (2.43)$$

R_l , t_l , R_r , t_r are rotation matrices and translation vectors describe how the left and right camera coordinates transformed from world coordinate system accordingly. From equation 2.42 and 2.43 we can derive that:

$$\begin{aligned} Q_r &= R_r R_l^{-1} (Q_l - t_l) + t_r \\ &= R_r R_l^{-1} (Q_l - t_l + R_l R_r^{-1} t_r) \\ &= R (Q_l - t) \end{aligned} \quad (2.44)$$

where R and t are the relative transformation between two camera coordinate systems and these are the extrinsic parameters we are looking for. Because the

2.4 Estimate extrinsic parameters from two images

stereo rig captures calibration grid from n different positions, for each position we have:

$$R_i = R_{ri} R_{li}^{-1} \quad (2.45)$$

$$T_i = T_{li} - R_{li} R_{ri}^{-1} T_{ri} \quad \text{with } i = 1 \dots n \quad (2.46)$$

Because the configuration of stereo rig does not change, R_i, T_i should be the same for all n different positions, but they are slightly different due to noise. In [64], the author suggests to perform final non-linear optimisation over all m calibration points at n different positions, by using the intrinsic parameters of the left and right cameras and the 6 extrinsic stereo parameters (R, t) together. After having intrinsic and extrinsic parameters, a stereo image pair can be rectified to the ideal model introduced in section 2.2.1. After rectification, the projections of a world point on the both images appear almost on same horizontal line.

2.4 Estimate extrinsic parameters from two images

Instead of trying to find extrinsic parameters using a checkerboard, Hartley [65] attempts to find the homographies (for left and right images) that map the epipoles to infinity, while minimizing the computed disparities between two images. As illustrated in Figure 2.8, the projections of a world 3D point P on to left and right image planes are denoted as p_l and p_r . Without loss of generality, we assume the world coordinate system is the same as the coordinate system of the left camera. We define P_l as the coordinate viewed from left camera and P_r as the coordinate of P viewed from right camera. Given the relative rotation matrix R and translation vector $\mathbf{t}$ between two cameras, the right camera is located at coordinate $\mathbf{t}$. Then the epipolar plane which connects P, O_l and O_r can be represented as:

$$(P_l - \mathbf{t})^T (\mathbf{t} \times P_l) = 0 \quad (2.47)$$

P_l and P_r has the relation $P_r = R(P_l - \mathbf{t})$. By substituting $(P_l - \mathbf{t})$ in equation 2.47, we get:

$$(R^{-1} P_r)^T (\mathbf{t} \times P_l) = (R^T P_r)^T (\mathbf{t} \times P_l) = 0 \quad (2.48)$$

2. BACKGROUND AND LITERATURE REVIEW

The cross product $\mathbf{t} \times P_l$ can be rewritten as dot product using skew symmetric matrix as:

$$\mathbf{t}P_l = [\mathbf{t}]_{\times}P_l \quad \text{and} \quad [\mathbf{t}]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (2.49)$$

By substituting the cross product in equation 2.48 we get:

$$P_r^T R[\mathbf{t}]_{\times} P_l = 0 \quad (2.50)$$

If we define $R[\mathbf{t}]_{\times} = E$, we get:

$$P_r^T E P_l = 0 \quad (2.51)$$

E is essential matrix and note that no intrinsic parameters are included in E which means it relates points in camera coordinate system not in pixel coordinate. In order to relate points in pixel coordinate, intrinsic parameters need to be introduced in the equation. With intrinsic parameters, we have $p = KP$, or

Figure 2.8: Epipolar geometry.

2.4 Estimate extrinsic parameters from two images

$P = K^{-1}p$ where p is the projection of P in pixel coordinate. Thus equation 2.51 becomes:

$$K_r^{-1}p_r EK_l^{-1}p_l = p_r^T (K_r^{-1})^T EK_l^{-1}p_l = 0 \quad (2.52)$$

If we define $F = (K_r^{-1})^T EK_l^{-1}$ we get:

$$p_r^T F p_l = 0 \quad (2.53)$$

F is the fundamental matrix which relates points in pixel coordinate and it is of rank 2.

2.4.1 Fundamental matrix estimation

Many methods to estimate the fundamental matrix have been proposed, and they can be mainly divided into two categories: linear methods and iterative methods (see [65] and [66] for details). Given left and right images, we find corresponding matching pairs $\mathbf{m} \leftrightarrow \mathbf{m}'$. Given n such matching pairs: $m_i = [x_i, y_i, 1]^T \leftrightarrow \mathbf{m}' = [x'_i, y'_i, 1]^T, i = 1, \dots, n$. If we rearrange F into a vector form $\mathbf{f}_{9 \times 1}$, a linear system can be built as:

$$A\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = 0 \quad (2.54)$$

Because F has 7 degrees of freedom, theoretically, if we have $n \geq 7$ pairs of matching points, we can have a solution for $\mathbf{f}$. However, in practice, due to measurement and computing error, an exact solution for equation 2.54 does not exist. An alternative solution is that we minimize $\|A\mathbf{f}\|$ subject to the condition $\|\mathbf{f}\| = 1$. 8 points correspondences are required in which the redundancy of points allows the minimization of the error in estimating F . The problem becomes:

$$\min_F \sum_i (m_i^T F m'_i)^2 \quad (2.55)$$

The solution can be obtained in closed form using SVD [67].

According to [68], the linear method for fundamental matrix estimation works better if the input data are properly normalized. The suggested normalization is a translation and scaling of the matching points such that the centroid of the

2. BACKGROUND AND LITERATURE REVIEW

feature points is the origin, and the root mean square distance of the feature points to the origin is $\sqrt{2}$. Given $n > 8$ matching pairs, the normalized eight point fundamental estimation approach is as follows.

Algorithm 1 normalized 8-point fundamental estimation algorithm

1. Normalization: transform the image coordinates according to $\hat{\mathbf{m}}_i = T\mathbf{m}_i$ and $\hat{\mathbf{m}}'_i = T'\mathbf{m}'_i$, where T and T' are the normalizing transformations.
 2. Find fundamental matrix $\hat{F}$ from the normalized matching $\hat{\mathbf{m}} \leftrightarrow \hat{\mathbf{m}}'$ as described in equation 2.55.
 3. Denormalization: The fundamental estimation result is given by: $F = (T')^T \hat{F} T$ corresponding to the original dataset.
-

Normally, the corresponding feature is obtained using descriptor similarity which cannot guaranteed to be noise free. Outliers will severely affect fundamental matrix estimation. In order to make the estimation robust, three methods are widely used in practice: M-estimator, Least-Median-Squares(LMedS) and RANSAC (See [65] for details). The M-estimator tries to reduce the effects of outliers by assigning less weight for outliers. The LMedS and RANSAC are similar. Both approaches randomly select a subset of point matchings to compute F . The difference is that: For each F , LMedS calculates the median distance between the epipolar lines and points. The goal of LMedS is to find a fundamental matrix that minimize this median. On the other hand, RANSAC calculates for each F by finding the one with maximum number of inliers. Once all the outliers are removed, F is recalculated with remaining inliers to obtain a better estimation. The RANSAC-based fundamental matrix estimation approach is as follow: If intrinsic parameters of both cameras are known, better fundamental estimation results can be obtained if we undistort images beforehand which will directly affects the stereo rectification.

2.4 Estimate extrinsic parameters from two images

Algorithm 2 RANSAC-based fundamental matrix estimation

1. Detect feature points and extract feature descriptors in each image.
 2. Compute feature points correspondences using descriptor similarity.
 3. Repeat for N samples:
 - (a) Select a random sample of 8 feature matches and compute the fundamental matrix F as described in normalized 8-point fundamental estimation algorithm.
 - (b) Count the number of inliers that consist with F and F with the most inliers retained.
 4. Re-estimate F using all inliers.
-

2.4.2 Extract extrinsic parameters from essential matrix

Given fundamental matrix F and intrinsic parameters of both cameras, essential matrix E can be obtained as:

$$E = K_r^T F K_l \quad (2.56)$$

The Singular Value Decomposition of E is:

$$E = U \text{diag}(1, 1, 0) V^T \quad (2.57)$$

where $\text{diag}(\cdot)$ represents a diagonal matrix. Then the following two solutions are possible for R :

$$R = UWV^T \quad (2.58)$$

$$R = UW^T V^T \quad (2.59)$$

where $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and for $\mathbf{t}$:

$$\mathbf{t} = \pm u_3 \quad (2.60)$$

2. BACKGROUND AND LITERATURE REVIEW

where u_3 is the third column of U . Combinations of rotations and translations give four possible solutions for configuration of stereo cameras, among which only one is feasible as shown in Figure 2.9. The way to determine the correct solution

Figure 2.9: Four possible solutions for essential matrix decomposition.

is to test whether any point derived from any point correspondence lies in front of both cameras. Assuming (u_l, v_l) and (u_r, v_r) are the projections of the same 3D point P into left and right cameras respectively. Let $(\hat{u}_l, \hat{v}_l)$ and $(\hat{u}_r, \hat{v}_r)$ denotes the corresponding projections in normalized image coordinates with unit focal length. Then the depth of P is:

$$z = \frac{(\mathbf{r}_1 - \hat{u}_r * \mathbf{r}_3) \cdot \mathbf{t}}{(\mathbf{r}_1 - \hat{u}_r * \mathbf{r}_3) [\hat{u}_r \quad \hat{v}_r \quad 1]^T} \quad (2.61)$$

Then we have:

$$P_l = \begin{bmatrix} \hat{u}_l z \\ \hat{v}_l z \\ z \end{bmatrix} \quad \text{and} \quad P_r = R^T(P_l - \mathbf{t}) \quad (2.62)$$

If the depth of P_l or P_r is not positive, then the corresponding R and $\mathbf{t}$ is degenerate.

2.5 Homography estimation

Without knowing the parameters of a stereo rig, the transformation between two images of the same planar scene can be estimated and it is called homography. The Euclidean coordinate of a $2D$ image point $(x_1, x_2)^T$ can be represented by a 3D homogeneous vector $\mathbf{x} = (x_1, x_2, x_3)^T$. The advantage of using a homogeneous representation is that: the equations for perspective projection to the image plane using Euclidean coordinates are non-linear, however, they are linear when expressed in homogeneous coordinates. An important property of homogeneous representation is that two homogeneous coordinates are equivalent if one is the scalar multiple of the other:

$$(x_1, x_2, 1)^T \equiv (\lambda x_1, \lambda x_2, \lambda)^T \quad (2.63)$$

where λ is a scalar and $\lambda \neq 0$. This equivalence makes the conversion between Euclidean coordinate and homogeneous coordinate invertible:

$$(\lambda x_1, \lambda x_2, \lambda)^T \rightarrow \left(\frac{\lambda x_1}{\lambda}, \frac{\lambda x_2}{\lambda}\right) = (x_1, x_2)^T \quad (2.64)$$

The homography is defined in [65] as an invertible mapping from one projective plane to another such that collinearity hold before and after mapping. The mapping of a point $\mathbf{x} \in \mathbb{R}^2$ is represented by $H\mathbf{x}$. H is the homography matrix which is a non-singular 3×3 matrix.

It should be noticed that the transformations described by H and λH are the same which means that H can be multiplied by any non-zero constant value λ while keeping the homography transformation property. Even H has 9 entries, it is a homogeneous matrix with only 8 DoF. For a given optimization problem, there are often several ways to parametrize it. Although $\mathbf{H}$ has only 8 DOF, we parametrize it using 9 parameters. As discussed in [69], it is not necessary or advisable to use 8-parametrization by removing scale factor because when using the minimal parametrization, it is more likely for the optimization process stuck

2. BACKGROUND AND LITERATURE REVIEW

in the local minimum. Also, in our problem (will be discussed in Chapter A), the number of parameters is usually much more than 9. A minimal parametrization of $\mathbf{H}$ has little help on efficiency of our approach. Hence, the total number of parameters need to be solved to estimate homography is $9 + n$ where n is the number of keypoint correspondences.

2.5.1 Algorithms for homography estimation

The Direct Linear Transformation (DLT) method is a linear algorithm to estimate a homography given $2D$ feature correspondences $x_i \leftrightarrow x_i'$. Since we use homogeneous coordinate to represent each point, the transformation is given by the equation:

$$\lambda \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (2.65)$$

where λ is a non-zero constant to make the equation equal on both sides.

Divide the first row and the second row of equation 2.65 by the third row we can get:

$$-h_1x - h_2y - h_3 + (h_7x + h_8y + h_9)x' = 0 \quad (2.66)$$

$$-h_4x - h_5y - h_6 + (h_7x + h_8y + h_9)y' = 0 \quad (2.67)$$

We can rewrite equations 2.66 and 2.67 to matrix form as:

$$A_i h = 0 \quad (2.68)$$

where $A_i = \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{bmatrix}$ and

$$h = [h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7 \ h_8 \ h_9]^T$$

2 independent equations can be obtained from each feature correspondences, hence, only 4 feature correspondences are required to solve H which has 8 DoF as long as no 3 points are collinear. Each feature correspondence can provide one matrix as A_i , given 4 correspondences, a single 8×9 matrix A can be obtained by stacking those 4 matrices together. The solution of h can be found by solving $Ah = 0$.

If more than four point correspondences $x_i \leftrightarrow x_i'$ are given, then the equation $Ah = 0$ which is derived from equation 2.68 is over-determined. If all feature

correspondences are error-free, one exact homogeneous solution can be obtained. However, in practice, there will always be some noise, the points and feature correspondences will not be error-free and no exact solution can be obtained. In order to estimate h , a cost function should be minimized.

Often more than 4 point correspondences with uncertainty of points can be obtained by feature matching. A suitable cost function is desired, by minimizing which, the homography can be estimated.

2.5.1.1 Algebraic error

The cost function made up of algebraic error is the simplest cost function, by minimizing which the homography can be solved, and it has the form: $\|Ah\|$. The constraint $\|Ah\| = 1$ is necessary to guaranty a non-zero solution of h . By using SVD (Singular Value Decomposition), the solution of h can be obtained by finding the unit singular vector corresponding to the smallest singular value of A .

2.5.1.2 Geometric error

The geometric error represents the pixel distance between a point on the image and its correspondence on the other transformed image. By assuming errors exist only in the transformed image, the geometric error of a correspondences $x_i \rightarrow x_i'$ can be represented as:

$$\sum_i d(x_i', Hx_i)^2 \quad (2.69)$$

where d is a function that measure the Euclidean image distance between two points. H is the homography.

In practice, errors exist in both images. In that case, the symmetric geometric error can be defined so that both the forward (H) and backward (H^{-1}) transformations are evaluated. The symmetric geometric error is defined as:

$$\sum_i d(x_i', Hx_i)^2 + d(x_i, H^{-1}x_i')^2 \quad (2.70)$$

By minimizing equation 2.70, the homography can be determined.

2. BACKGROUND AND LITERATURE REVIEW

2.5.1.3 Reprojection error

In order to estimate the homography $\hat{H}$ as well as the point positions of $\hat{x}_i$ and $\hat{x}'_i$, the reprojection error function is defined as:

$$\sum_i d(x'_i, \hat{x}'_i)^2 + d(x_i, \hat{x}_i)^2 \quad (2.71)$$

where $\hat{x}'_i = \hat{H}\hat{x}_i$. It is more difficult to minimize this cost function compared to the cost function using geometric error. The difficulty is that it requires a simultaneous minimization over all choices of points $\hat{x}_i$ as well as the entries of the homography H . If n pairs of correspondences are chosen, the total parameters need to be minimized is $2n + 9$.

2.5.2 Robust homography estimation

The homography estimation method discussed above require correct feature correspondences as input. However, in practice, other sources of noise can "break down the algorithm", such as false correspondences, which means false matchings of feature points are given as input of the algorithm. In order to have better estimation of the homography, outliers need to be identified. Two common ways of dealing with such undesired false correspondences are discussed in this section.

2.5.2.1 RANSAC

When dealing with outliers, Random Sample Consensus (RANSAC) is one of the most popular outlier removal approach as in [70]. In the case of homography estimation, RANSAC work as follows: 4 feature correspondences are selected randomly and homography H can be obtained from those 4 correspondences. For all other correspondences, each of them can be determined as an inlier or outlier depending on whether the correspondence agrees with the estimated H . After a predefined number of iterations are completed, H that resulted in the biggest number of inliers is selected. Given all the inliers, H can be computed again as final refinement step.

In practice, it is impossible to try all combinations of correspondences. Hence, in order to apply RANSAC, one needs to determine the termination criteria that

is how many iterations is required before the algorithm terminate. Assume the total iteration of required for RANSAC is N . Given a probability of p that at least in one iteration, that randomly selected correspondences are noise free. Hartley and Zisserman showed in [65] that $N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)}$, where ϵ is the portion of outliers. s is the number of data that need to be sampled in each iteration.

2.5.2.2 Least Median of Squares

Besides RANSAC, there are many research works trying to dealing with outliers. Instead of using squared distance, we can use absolute value of the distance. By doing this, the robustness of the algorithm can be increased because the outliers will be penalized less compared to the square error. An alternative to deal with outliers is to uses median of the squared residuals to replace the sum in the cost function as discussed in [71]. This method is called Least Median of Squares (LMS). The pros and cons of LMS is that (quoted from [65]):

It requires no setting of thresholds or a priori knowledge of the variance of the error. The disadvantage of LMS is that it fails if more than half the data is outlying, for then the median distance will be to an outlier. The solution is to use the proportion of outliers to determine the selection distance. For example if there are 50% outliers then a distance below the median value should be used.

2.6 Visual odometry estimation

Most of the early research in visual odometry ([72], [73], [74], [75], [76]) was inspired by the Mars exploration program to give planetary rovers 6-DoF motion estimation. One of the first works in visual odometry is attributed to Moravec [75]. In 1976 he successfully implemented a visual odometry system on the Stanford Cart to perform corrections on the control of the vehicle autonomously. In that technical report, the basic concepts of tracking visual features and depth estimation were introduced. Besides [75], one of Moravec's works, which is known

2. BACKGROUND AND LITERATURE REVIEW

as the Moravec corner detector [77], is the predecessor of some other famous corner detectors such as Förstner [78] and Harris [79]. A variety of approaches for visual odometry tasks have been introduced in the past. Based on the sensor type these approaches can be divided into two categories, namely monocular camera odometry and stereo camera odometry.

Monocular visual odometry uses only single camera. However, the motion can only be estimated up to a scaling factor. Compared to monocular case, stereo visual odometry has the advantage that the 3D coordinates of the features relative to the stereo rig are directly calculated by triangulation at each camera pose and used to determine the relative transformation. Hence, the scale ambiguity does not exist in stereo case. However, if the depth of the scene is too big compared to the stereo baseline (the distance between two mono cameras), the triangulation fails to get reasonable 3D coordinates of features. As a result, the stereo visual odometry would become ineffective and a monocular one has to be used. Although the focus of our research is stereo visual odometry, for the purpose of completeness, the progress of both types of visual odometry will be discussed in this section.

2.6.1 Monocular visual odometry

In the case of monocular vision, since the absolute scale is unknown, both of the relative transformation and the 3D structure of the scene have to be estimated from 2D bearing measurements. Normally, the distance between the two mono cameras is set to one when at the starting point. The following camera poses can be estimated relative to the first two poses.

During the last decades, successful implementations of monocular visual odometry using either perspective or omnidirectional cameras have been obtained. Related works can be divided into two categories: appearance based methods and feature based methods.

In the category of appearance based methods, in [80], authors used a planar surface to compute optical flow. The equation of the planar surface (such as a road) should agree with the camera motion. In [81], authors used optical flow computed by the Lucas-Kanade method [17] to estimate camera motion. In order to save computational power, the authors compute only a limited number of flow vectors, instead of using the whole image to compute the optical flow. They use Monte Carlo sampling to generate sample points for optical flow computation. Instead of optical flow, authors in [82] presented a direct approach. Feature detectors can be designed to detect interesting feature points in an image. However, the detected feature or the feature descriptor may not be stable. For example, two corresponding features detected in two images may not reflect the same 3D world point (as they should). In order to cope with unreliable feature points, in direct methods, all pixels of the images are taken into account to estimate camera motion and no computation for the feature tracking or optical flow is required. In [83] authors used Fourier-Mellin transform to register images captured from multiple perspective views of the ground plane. In [84], authors proposed a method to estimate approximate translational and rotational velocity information from a monocular camera, using template tracking on the center of the scene.

In the feature based methods, the first real-time, large scale monocular visual odometry method was introduced in [7]. The contribution of their paper was to use a five-point approach [85] to estimate the camera motion using RANSAC manner. Corke et al. [86] used omnidirectional imagery to estimate the motion of planetary rover using robust estimation of optical flow and subsequent integration of the flow. In [87] and [88], authors presented approaches based on local bundle adjustment (bundle adjustment only applied on the past few frames) to recover both of the camera motion and 3D scene.

2. BACKGROUND AND LITERATURE REVIEW

2.6.2 Stereo visual odometry

The stereo visual odometry means that the 3D coordinates of features are calculated by direct triangulation at each camera position and used to estimate the relative motion of camera. In [89] a binocular camera is used for tracking corners. The motion estimation accuracy is improved by properly modeling the error covariance of triangulated points. Olson et al. [76] introduced an absolute orientation sensor and used Förstner corner detector. It is shown that when using an absolute orientation sensor, the error growth can be bounded as a linear function of the distance travelled.

A similar stereo visual odometry approach was presented by Lacroix et al. [72]. However, they observed that due to the noise in images, the detected keypoints often drifted which means after some time, the tracked keypoints no longer correspond to the original pixels. Hence, instead of using a corner detector, the candidate keypoints that needed to be tracked are selected using a correlation score computed from the pixel of keypoint together with its neighbors. The error of the calculated disparity can be computed using dense stereo.

Cheng and Maimone et al. described their visual odometry implementation on the Mars rovers in [90] and [91]. After feature detection using the Harris corner detector, they used the correlation function around features to model the error covariance of the image points. When estimating the motion, they used random sample consensus as proposed by Nister [7]. In [92], authors used the Shi-Tomasi corner detector [93] to detect feature points. They eliminated those points that have low confidence depending on the disparity map. Different from previous works, besides the motion estimation using least squares, they used iterative closest point (ICP) [94] to refine the pose estimation.

The works mentioned above have been designed for 3D-to-3D points registration which means the 3D coordinates of features are triangu-

lated using stereo matching at each camera location and the relative motion is estimated by registering these 3D points. Nister et al. [7] presented a completely different approach which is known as the first real-time and long-run visual odometry implementation. Firstly, unlike other works, instead of tracking features across several frames, they detected corners in each frame and matched them with previous frames. In this way, the feature drift in the tracking approaches can be ignored. Secondly, instead of computing the camera motion based on the 3D-to-3D points registration, they used a 3D-to-2D approach by minimizing the reprojection error. A more accurate motion estimation can be obtained through 3D-to-2D approach because the error introduced by triangulating 3D points of stereo pairs can be eliminated.

2.7 Motion estimation for stereo visual odometry

The study of stereo vision allow us to sensing the surrounding environments in 3D space which permits an estimation of cameras' position in unknown environment. Although many steps such as feature detection or feature matching are involved in visual odometry estimation, the core computation step is the motion estimation which calculates the camera motion between current frame and the previous frame. This section introduces the approaches to compute the camera motion given feature correspondences. Generally, depending on whether the features are triangulated to get 3D coordinates, the approaches can be divided into three categories:

- *2D-to-2D*: Only 2D coordinates available for features in the current and previous frames.
- *3D-to-3D*: 3D coordinates are available for both current and previous frames.

2. BACKGROUND AND LITERATURE REVIEW

- *3D-to-2D*: 3D coordinates are available for features in the previous frame but only 2D coordinates available for current frame.

In this thesis, the *3D-to-3D* approach is used in Chapter 3 and the *3D-to-2D* approach is used in Chapter 4. The *2D-to-2D* approach is also listed here for complementary purpose.

2.7.1 2D-to-2D motion estimation

In this case, camera motion is estimated from 2D feature correspondences between two consecutive frames. Given a calibrated camera, the motion between two frames can be described by essential matrix E . E can be calculated by 2D feature correspondences using the epipolar constraint. As illustrated in Figure 2.10, this constraint determines the lines on which the corresponding features P_L and P_R lie which can be formulated as $P_R^T E P_L = 0$. The camera motion parameters are included in E as:

$$E = R[t]_x \quad (2.72)$$

in which R and t are 3×3 rotation matrix and translation vector respectively, and

$$[t]_x \simeq \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (2.73)$$

where $\simeq$ denotes the equation is up to a scale factor. A robust 8-point algorithm has been described in section 2.4.1 which involves no more than the construction and solution of a set of linear equations.

After obtaining essential matrix E , the translation vector and rotation matrix can be extracted from it using SVD as described in section 2.4.2. The rotation matrix can be extracted from the essential matrix properly while the extracted translation vector has an unknown scaling factor. Although the absolute scale of translation cannot be computed from two frames, a relative scale can be determined with

Figure 2.10: Epipolar geometry.

respect to the translation between the previous image pairs. Given X_{k-1} and X_k as the triangulated 3D points from two image pairs, the relative scale s can be calculated as the distance ratio between the point pairs in X_{k-1} and X_k .

$$s = \frac{\|X_{k-1,i} - X_{k-1,j}\|}{\|X_{k,i} - X_{k,j}\|} \quad (2.74)$$

This distance ratio is normally calculated several times using different point pairs and averaged to reduce the effects of noise. The extracted translation vector is then scaled by s to obtain the relative translation.

2.7.2 3D-to-3D motion estimation

If 3D coordinates of features are available, the camera motion can be determined by aligning the corresponding 3D points of image features in consecutive frames (in chapter 3). The transformation matrix T_k can be obtained by minimizing the euclidean distance between corresponding 3D feature points.

$$\arg \min_{T_k} \sum_i \|X_k^i - T_k X_{k-1}^i\|^2 \quad (2.75)$$

2. BACKGROUND AND LITERATURE REVIEW

where X_k^i denotes the i th 3D feature point in frame k . The minimal case to solve 2.75 requires three 3D-to-3D non-collinear correspondences. For the case of more than three correspondences, we can compute the relative transformation in closed form. The centers of the two point sets are

$$\bar{X}_{k-1} = \frac{1}{N} \sum_{i=1}^N X_{k-1}^i \quad \text{and} \quad \bar{X}_k = \frac{1}{N} \sum_{i=1}^N X_k^i$$

where N is the total number of correspondences. Before computing the transformation, we move two centers of point clouds to the same point by subtracting the corresponding center of point clouds from each point in the two point sets:

$$\begin{aligned} \hat{X}_{k-1} &= \{X_{k-1}^i - \bar{X}_{k-1}\} = \{\hat{X}_{k-1}^i\} \\ \hat{X}_k &= \{X_k^i - \bar{X}_k\} = \{\hat{X}_k^i\} \end{aligned}$$

Let $W = \sum_{i=1}^N \hat{X}_{k-1}^i \hat{X}_k^{iT}$, the *SVD* of W can be denoted by $W = U \text{diag}(\sigma_1, \sigma_2, \sigma_3) V^T$ where $U, V \in \mathbb{R}^{3 \times 3}$ are unitary and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are the singular values of W . Then the rotation matrix and translation vector can be obtained as:

$$\begin{aligned} R &= UV^T \\ t &= \hat{X}_k - R\hat{X}_{k-1} \end{aligned}$$

The computed transformation has absolute scale and the current position of the camera can be calculated by concatenating the transformations between consecutive frames.

2.7.3 3D-to-2D motion estimation

In this case, although 3D coordinates for image features are available, transformation of camera is calculated from the 3D-to-2D correspondences (will be used in chapter 4). As pointed out in [7], 3D-to-2D

motion estimation is more accurate than 3D-to-3D case by minimizing the reprojection error:

$$\arg \min_{T_k} \sum_i \|p_k^i - \hat{p}_{k-1}^i\|^2 \quad (2.76)$$

where $\hat{p}_{k-1}^i$ is the projection of the 3D point X_{k-1}^i on image I_k using transformation T_k .

A simple and straight forward solution is the direct linear transformation [69]. For each correspondence $X_{k-1}^i \rightarrow p_k^i$, we can derive a equation

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -x & -y & -z & -1 & xv & yv & zv & v \\ x & y & z & 1 & 0 & 0 & 0 & 0 & -xu & -yu & -zu & -u \end{bmatrix} \begin{bmatrix} P^1 \\ P^2 \\ P^3 \end{bmatrix} = 0 \quad (2.77)$$

where P^i is a 4 dimensional vector and it is the i th row of camera matrix P such that $p_k^i = PX_{k-1}^i$. Since the matrix P is 3×4 and has 11-DOF (ignoring scale), at least 11 equations are required to solve for P . Each 3D-to-2D correspondence provides two equations, at least 6 pairs of correspondences are required to solve for P . $AP = 0$ is generated by stacking 6 correspondences. By using SVD, P can be solved from A . Since $P = [R|t]$ the rotation matrix and translation vector can be decomposed from P . Nonlinear optimization methods such as Gauss-Newton or Levenberg-Marquardt can be used to further refine R and t .

2.7.4 Pose estimation given camera motion

The camera motion can be estimated from two consecutive frames. The camera motion is a rigid transformation including rotation and translation in 3D. A rotation R and a translation t is called a pose. Such pose transformation can be expressed as:

$$x_b = R_{ba}x_a + t_{ba} \quad (2.78)$$

2. BACKGROUND AND LITERATURE REVIEW

Figure 2.11: Rigid transformation between three different frames a, b and c. The 6 DoF frames are represented using three coordinate axis which can be rotated in 3D.

As shown in Figure 2.11, the transformation $R_{ba}x_a + t_{ba}$ transforms a point x_a in frame a to a point x_b in frame b . This transformation can be written as a 4×4 matrix:

$$T_{ba} = \begin{bmatrix} R_{ba} & t_{ba} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (2.79)$$

The current pose is the concatenation of multiple camera rigid motion. In Figure 2.11 the current pose of c can be represented by concatenation of the transformation from a to b and then from b to c which are performed using matrix multiplication:

$$\begin{aligned} \begin{bmatrix} R_{ca} & t_{ca} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} &= \begin{bmatrix} R_{cb} & t_{cb} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} R_{ba} & t_{ba} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_{cb}R_{ba} & R_{cb}t_{cb} + t_{cb} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \end{aligned} \quad (2.80)$$

Given a transformation T_{ba} , the inverse transformation T_{ab} can be computed using matrix inverse:

$$T_{ab} = T_{ba}^{-1} = \begin{bmatrix} R_{ba}^T & -R_{ba}^T t_{ba} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (2.81)$$

2.7.5 FOVIS: a stereo visual odometry estimation approach

The visual odometry estimation approach (*FOVIS*) in [14] follows a standard stereo visual odometry pipeline and achieves better performance than many of the existing approaches and we compare with our proposed stereo visual odometry with FOVIS in chapter 4. The details of the algorithm is described as follows:

1. *Image preprocessing*: Each captured image is smoothed with a Gaussian kernel and Gaussian pyramid is constructed to make the detected feature to be scale-invariant.
2. *Feature detection*: FAST [95] features are detected at each level of Gaussian pyramid. Those features whose depth cannot be obtained are discarded. The whole image is divided into 80×80 buckets and 25 features in each bucket are retained to make the features distribute more uniformly in the image.
3. *Initial rotation estimation*: A proper estimation of the rotation helps to constrain the searching windows when finding corresponding features between two consecutive frames. The technique to estimate this initial rotation is to directly minimize the sum of squared pixel errors between two consecutive frames as proposed in [96].
4. *Feature matching*: The feature descriptor is created by using the intensity value of 9×9 square patch around the feature. All intensity values are normalized to zero mean and each intensity in the patch is represented by 1 byte which makes the descriptor more suitable for SIMD instructions. The search for the corresponding feature is constrained within a window defined by the initial rotation estimation. A feature matching is accepted only if two features have the lowest SAD (sum of absolute difference) to each other.

2. BACKGROUND AND LITERATURE REVIEW

5. *Inlier detection* A graph of consistent feature matches is constructed as proposed in [97]. With the fact that the Euclidean distance between two features (at one time) should be consistent with their distance at another time. The vertex in the graph is a matched feature pair. An edge between two vertex is formed if the 3D distance between the features stays the same between two consecutive frames. Hence the inlier finding problem becomes of that finding the maximal clique in the graph.
6. *Motion estimation* The relative motion between two consecutive frames is estimated by minimizing the reprojection error between matched features. To reduce the drifting problem of visual odometry, a keyframe technique is applied. Instead of comparing new frame with previous frame, a keyframe is created to compare with the new frame. If sufficient inliers can be obtained between new frame and keyframe, then the keyframe remains unchanged. Otherwise, the keyframe is replace by the new frame.

2.8 Cost function minimization methods

Given the aforementioned non-linear cost function, an iterative minimization approach is desired to obtain camera motions (for both homography and visual odometry) between frames. Compare to the direct approach, the iterative approach can have more accurate results. However, they are tend to be slower than the direct method and may not converge. In this section, three most common iterative minimization methods are discussed, namely *Gauss-Newton* method, *Graident Descent* and *Levenberg-Marquardt*.

2.8.1 Newton iteration

To illustrate the problem, assume we are given a non-linear function $X = f(P)$ where $X \in \mathbb{R}^n$ is the measurement vector and $P \in \mathbb{R}^m$ is

the parameter vector. In practice, with the presence of noise, it is often impossible to find an exact solution of P that satisfies the function. Alternatively, we seek the vector $\hat{P}$ that satisfies $X = f(\hat{P}) - \epsilon$ where $\|\epsilon\|$ is minimized. In other words, we want to minimize the function $\epsilon = X - f(\hat{p})$. To solve for P , we start with an initial estimation of P_0 and under the assumption that f is locally linear, the estimation of P_i is updated iteratively until convergence. By assuming f is locally linear we can have:

$$f(P_i + \Delta_i) = f(P_i) + J_i \Delta_i \quad (2.82)$$

where J_i is the Jacobian $J = \partial f / \partial P$ at P_i . We want to find a point $f(P_{i+1})$, with $P_{i+1} = P_0 + \Delta$, which minimizes function:

$$\begin{aligned} f(P_{i+1}) - X &= f(P_i) + J_i \Delta_i - X \\ &= \epsilon_i + J_i \Delta_i \end{aligned} \quad (2.83)$$

where ϵ_i is the error at i th iteration. So we want to minimize $\|\epsilon_i + J_i \Delta_i\|$ for some vector Δ_i which is a linear minimization problem and Δ_i can be obtained by solving the normal equation:

$$J_i^T J_i \Delta_i = -J_i^T \epsilon_i \quad (2.84)$$

or by using the pseudo-inverse:

$$\Delta_i = -J_i^+ \epsilon_i \quad (2.85)$$

thus we can update estimation of P as:

$$P_{i+1} = P_i + \Delta_i \quad (2.86)$$

Ideally, the iteration will continue until the algorithm find the optimal solution $\hat{P}$. Unfortunately, sometimes the iteration converges at a local minimal or does not converge at all. A proper initial guess can reduce the risk that the algorithm stuck at local minimal.

2. BACKGROUND AND LITERATURE REVIEW

2.8.2 Gauss-Newton method

The Gauss-Newton method is a well-known technique for solving non-linear least square problem. Suppose the cost function we want to minimize is:

$$\begin{aligned} g(P) &= \frac{1}{2} \|\epsilon(P)\|^2 \\ &= \frac{1}{2} \epsilon(P)^T \epsilon(P) \end{aligned} \quad (2.87)$$

where $\epsilon(P) = f(P) - X$. Expand equation 2.87 in a Taylor series up to second degree we can get:

$$g(P + \Delta) = g + g_{\mathbf{P}} \Delta + \Delta^T g_{\mathbf{P}\mathbf{P}} \Delta / 2 \quad (2.88)$$

where the subscript $\mathbf{P}$ denotes differentiation. Differentiating equation 2.88 with respect to Δ and setting it to zero we can get:

$$g_{\mathbf{P}\mathbf{P}} \Delta = -g_{\mathbf{P}} \quad (2.89)$$

where $g_{\mathbf{P}\mathbf{P}}$ is the matrix of second derivatives, the Hessian of g . the (i, j) th entry of $g_{\mathbf{P}\mathbf{P}}$ is $\partial^2 g / \partial p_i \partial p_j$ and p_i and p_j are the i th and j th parameters. For equation 2.87, gradient vector $g_{\mathbf{P}}$ can be computed as $\epsilon_{\mathbf{P}}^T \epsilon$. We can rewrite $\epsilon_{\mathbf{P}} = f_{\mathbf{P}} = J$. Hence, $g_{\mathbf{P}} = J^T \epsilon$. The Hessian can be computed by differentiating $g_{\mathbf{P}} = \epsilon_{\mathbf{P}}^T \epsilon$ the second time:

$$g_{\mathbf{P}\mathbf{P}} = \epsilon_{\mathbf{P}}^T \epsilon_{\mathbf{P}} + \epsilon_{\mathbf{P}\mathbf{P}}^T \epsilon \quad (2.90)$$

By assuming $f(P)$ is linear, equation 2.90 can be simplified to:

$$g_{\mathbf{P}\mathbf{P}} = \epsilon_{\mathbf{P}}^T \epsilon_{\mathbf{P}} = J^T J \quad (2.91)$$

This iterative method that uses $J^T J$ to approximate the Hessian is known as *Gauss-Newton* method.

Figure 2.12: Slow convergence due to zig-zaging property of gradient descent.

2.8.3 Gradient descent

In order to speed up the iterative procedure, we want to decrease the cost function most rapidly in each iteration. The negative gradient vector $-g_{\mathbf{P}} = \epsilon_{\mathbf{P}}^T \epsilon$ defines this most rapid decrease direction. The procedure where we move g along this negative gradient direction is known as *gradient descent*. The moving step can be estimated by searching along this direction to find function minimum. The parameter increment Δ is calculated from:

$$\lambda \Delta = -g_{\mathbf{P}} \quad (2.92)$$

where λ is the length of the step. Although gradient descent update P towards function minimum in a locally most rapid way, it may overshoot the minimum and needs to move back. This will cause the slow convergence due to zig-zagging as shown in Figure 2.12.

By comparing equation 2.89 and equation 2.92, we can see that in gradient descent, the Hessian is approximated by a scalar matrix λI while in Gauss-Newton, the Hessian is approximated by $J^T J$.

2. BACKGROUND AND LITERATURE REVIEW

2.8.4 Levenberg-Marquardt method

Sometimes gradient descent is not a good option to solve a non-linear minimization problem due to zig-zagging problem. However, it can be combined with the Gauss-Newton method to speed up convergence. This combination introduces a new iterative method which is called the *Levenberg-Marquardt* method.

In the Levenberg-Marquardt method, the normal equation $J^T J \Delta = -J^T \epsilon$ is replaced by:

$$(J^T J \Delta + \lambda I) = -J^T \epsilon \quad (2.93)$$

λ is called the damping parameter. A carefully chosen λ can guarantee a rapid convergence of the algorithm, otherwise, the algorithm may suffer from zig-zagging property of gradient descent. Marquardt suggested to give the damping parameter an initial value λ_0 , and a converging factor $v > 1$. In each iteration, if a Δ that can decrease the error function is found, the result is updated by Δ and λ is divided by v . Otherwise, λ is multiplied by v and the whole process is repeated again until a proper Δ is found that can decrease error function. If λ is very small, equation 2.93 is close to the Gauss-Newton method. When λ is big, the left side of equation 2.93 is dominated by λI which is essentially gradient descent method. Hence when close to the solution, the Gauss-Newton method will be prominent which will lead to rapid convergence. In other cases, gradient descent will take the lead and guarantee a decrease in the cost function when the convergence is difficult.

2.9 Conclusion

In this chapter, basic concepts of camera calibration and camera transformation are introduced. A study of the most commonly used camera

calibrating methods of the last few decades are presented. These techniques cover a wide range of the classical hard calibration of image sensors which begin from a previous knowledge of a set of 3D points and their corresponding 2D projections on an image plane in order to estimate both intrinsic and extrinsic camera parameters. Different cost functions for camera motion estimation and techniques to minimize these cost functions are introduced. Recent stereo visual odometry approach is described and will be used as benchmark to compare with in chapter 4. Commonly used techniques to solve non-linear minimization problem are also introduced.

Chapter 3

Motion Estimation on ceiling vision

In this chapter, we describe our approaches for 6-DOF visual odometry estimation on ceiling vision using a camera only. There is no single visual odometry solution that is ideal for every possible environment. Compared to a outdoor environment, an indoor environment has no infrared light, so we choose Kinect as sensor for motion estimation purposes. Two types of features, namely keypoint features and edge features will be discussed and the visual odometry results using these two types of features will be shown in this chapter.

We mounted the Kinect on the robot and pointed it up to the ceiling as in Figure 3.1. Ceiling vision has an advantage of no or little obstruction, as well as only involving rotation and affine transformation (without scale changes) compared to the frontal view. Also in ceiling vision, the field of view of the camera is not likely to be filled with non-stationary objects (such as moving people), which cannot be used as reference points to determine the visual odometry. Calibration and rectification are performed to achieve undistorted image frames. In this chapter, one point feature-based approach and one edge feature-based approach with principal direction approach are introduced.

3. MOTION ESTIMATION ON CEILING VISION

Figure 3.1: Kinect mounted on iRobot Create platform.

3.1 Keypoint feature-based visual odometry

We first used point feature-based approach for visual odometry estimation as described in [98]. Keypoints are detected and matched from two consecutive images. In chapter 4, “FAST” detector is used to extract features from images for efficiency consideration. Unlike front-view images, orientation of the ceiling images change a lot when the robot makes a turn. If the camera is rotated between two consecutive frames, the pixel configuration of a corner may change significantly, which makes the “FAST” detector not very consistent for rotation changes. Several other popular feature detectors including Harris corner [79], SIFT [99] and SURF [100], have been tested. However, the Harris corner detector lacks a good descriptor that hinders feature matching between frames of more than 100 features each. SIFT and SURF, on the other hand, have robust 128-dimension and 64-dimension descriptors which are distinctive and can provide good matching results. SURF has the advantage of speed, which has been

claimed to be several times faster and more robust against image transformation than SIFT. Therefore SURF is chosen for calculation of visual odometry. First, SURF features from the camera images are extracted and localized in 3D space using the corresponding depth images. We then match these features between every pair of acquired images. After that RANSAC is applied to eliminate the outliers (false matching). The rotation and translation between them can be estimated effectively using Singular Value Decomposition across all inliers. To achieve real-time processing, we match the current image only (versus the previous image).

3.1.1 Estimation of ego motion

The Kinect camera is mounted on top of the robot, pointing up to the ceiling. RGB and depth images are retrieved from the sensor. As the two images are provided by independent cameras (vision and range), calibration and rectification [101] are needed to align these two images and to achieve undistorted image frames.

SURF features are detected and extracted on each frame and matched with those of the previous frame. Matching approach is based on the kd-tree algorithm, targeting the 128-dimension descriptors of each SURF feature. Naturally, false matches occur. These false matches (outliers) are filtered out by applying Random Sample Consensus (RANSAC) with a 3D affine transformation model:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T$$

where (x, y, z) and (x', y', z') indicate the 3D coordinates of a feature point in the previous frame and its correspondence in the current frame. R and T are 3×3 rotation matrix and a 3×1 translation vector respectively. For each SURF feature (in 2D pixel coordinate) in the current frame, its initial matching can be established by finding

3. MOTION ESTIMATION ON CEILING VISION

(a) Initial matchings indicated by lines.

(b) After RANSAC: Outliers (false matches) removed; inliers indicated by yellow lines.

Figure 3.2: SURF features matching between 2 frames corresponding to 2 consecutive view of the robot (blue lines indicates frame boundary).

3.1 Keypoint feature-based visual odometry

the nearest neighbor (closest Euclidean distance in descriptor space) in the previous frame. A KD-tree is used here to speed up this process. As can be seen in Figure 3.2(a), some of the features between two frames are falsely matched. To eliminate these outliers, the location of SURF features are translated into 3D real world coordinate (x, y, z) . A subset of 3 pairs from the initial matching pairs are selected randomly. The rotation matrix R and the translation vector T can be calculated with the 3D coordinates of these 3 matching pairs using Singular Value Decomposition (SVD) [102]. Let $W = \sum_{i=1}^n p_i q_i^T$, where q_i denotes the 3D coordinate of the feature point in current frame and p_i is the 3D coordinate of the corresponding feature point in previous frame. n is the number of matching pairs that have been chosen (here is 3). The SVD of W can be expressed as:

$$W = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T$$

where $U, V \in \mathbb{R}^{3 \times 3}$ are unitary and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are the singular values of W . R and T can be calculated as:

$$R = UV^T \quad \text{and} \quad T = \mu_q - R\mu_p$$

where $\mu_p = \frac{1}{n} \sum_{i=1}^n p_i$ and $\mu_q = \frac{1}{n} \sum_{i=1}^n q_i$.

R and T are then used for calculating the re-projection error e of the remaining pairs of correspondence. The re-projection error is defined as the Euclidean distance between the transformed point of point A in the previous frame according to the model $R \times A + T$ and the original correspondence of A in the current frame:

$$e = \left\| \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} - R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T \right\|$$

The re-projection error threshold T_{thres} is determined such that among all the initial matches, only those pairs with $e < T_{thres}$ will be regarded as inliers and kept. We assume that the measurement error

3. MOTION ESTIMATION ON CEILING VISION

is Gaussian distribution with zero mean and standard deviation σ . Then the value of T_{thres} can be calculated. We would like to choose a re-projection error e such that with a probability α the matching is an inlier [69]. In this case e^2 is a sum of squared Gaussian variables and follows a Chi-square distribution χ_m^2 with m degrees of freedom, where m equals the co-dimension of the model, which is 3 in our case. Knowing that the probability that the value of a χ_m^2 random variable is less than k^2 is given by the cumulative Chi-squared distribution, $F_m(k^2) = \int_0^{k^2} \chi_m^2(\varepsilon) d\varepsilon$, we can determine the inlier as following:

$$\begin{cases} \text{inlier} & \text{if } e^2 < T_{thres}^2 \\ \text{outlier} & \text{if } e^2 \geq T_{thres}^2 \end{cases} \text{ with } T_{thres}^2 = F_m^{-1}(\alpha)\sigma^2$$

Usually α is chosen as 0.95, so that there is a 95% probability that the point is an inlier (an inlier will only be incorrectly rejected 5% of the time) with $\alpha = 0.95$, $T_{thres}^2 = 7.81\sigma^2$.

The process is repeated for different random subsets, resulting in different transformations, with the number of inliers recorded for each transformation. The transformation supported by the highest number of inliers is selected to be the winning transformation. In the final step, all inliers from the winning transformation are used to re-estimate the transformation using SVD. The resulting transformation will be considered as the final relative transformation between these two consecutive frames. The current robot location and orientation in the global coordinate can be retrieved by accumulating the relative rotation and translation matrix.

3.1.2 Experimental Results of keypoint-based of visual odometry

The experiments were carried out using iCreate robot platform as in Figure 3.1. The Kinect camera was mounted on it and pointed up to the ceiling (as shown in Figure 3). The robot was manually controlled

3.1 Keypoint feature-based visual odometry

Figure 3.3: Visual odometry estimation results in RRC.

with a wired joystick and testing data was collected by steering the robot in the Robotic Research Center of Nanyang Technological University. During our test, the robot traversed a total distance of about 45 meters. The average number of frames per second for feature-matching based visual odometry algorithm is about 3. The detection of SURF features and the extraction of the descriptors need about 0.16 sec for each frame. It takes about 0.07 sec for finding the initial matching and 0.08 sec for RANSAC process.

In each frame, around 1000 SURF features can be detected and more than 100 pairs of SURF features can be matched by finding the nearest neighbor. After applying RANSAC algorithm, about 30% – 60% of the initial matching pairs will be confirmed as inliers and used to calculate the relative transformation between two consecutive frames.

The resulting visual odometry is shown in Figure 3.3. The robot started and ended at the same point. The blue dotted path is the ground truth generated using laser, wheel odometry and IMU. The red path is the estimated visual odometry using feature-matching based approach. As demonstrated in Figure 3.3, the rotation and translation errors are accumulated overtime. The second limitation is that the

3. MOTION ESTIMATION ON CEILING VISION

processing speed is about 0.3 second for each frame which is too slow for real-time applications. To deal with these limitations, in the next section, we will introduce an efficient and improved method that uses features on ceiling as a global reference to reduce the drift of visual odometry.

3.2 Visual odometry using principal direction

SURF features detection and matching as described in the previous section is time consuming for a real-time system. In this section, we explore another visual odometry approach on ceiling vision which can overcome the speed limitation. Given pointclouds obtained by Kinect, visual odometry can be obtained by registering point clouds from various time instants into one reference coordinate system to obtain 6 DoF motion parameters. The common method for point cloud registration is ICP. The general idea is to iteratively assign correspondences between the points of the two point clouds and to update motion parameters until the system converges. However, ICP is prone to local minimum as the registration tends to reinforce a possibly sub-optimal initial point correspondence. Alternatively, monocular images can be used to extract key-points and match them with previous frames using descriptor matching to roughly align two point clouds [103]. The correct data correspondences of ICP are much more likely to be found in pre-aligned point clouds. As a result, the registration performance can be improved. However, the runtime for key points detection and descriptors extraction is unable to reach real-time performance without using a graphics processing unit (GPU). Moreover, descriptor based methods suffer from the drawback that they cannot distinguish repeated patterns.

3.2 Visual odometry using principal direction

All frame-to-frame based algorithms suffer from error accumulation problem. Although the motion estimation error between two consecutive frames can be small enough, this error accumulates over time and leads to a bad estimation of robot position after a long run. The accumulated error can only be corrected by a loop closing technique. However, besides additional computation power and sensor requirements, loop closure detection can introduce new error and result in an even worse position estimation.

The main contribution of which edge-based visual odometry approach is that it can perform real-time principal direction detection that acts as a global reference to reduce accumulated error in frame-to-frame based methods. Also, it can perform well with repeated patterns since the direction detection is not completely based on local intensity changes. In many buildings, ceilings contain vertical and horizontal lines, and the orientation of these lines can be used as a global reference for robot's orientation. Even in the case where no typical lines or markings on the ceiling, the edges between walls and ceiling can also be utilised as the principal direction.

The main steps for the proposed principal direction detection approach are as follows. First, local linear features are extracted from monocular images grabbed by the Kinect camera pointing up to the ceiling. Second, the ceiling plane is detected using depth map obtained by the Kinect, and local features not belonging to the ceiling are eliminated to improve the accuracy. Then *Hough transform* is performed to extract lines and the principal direction is determined by a voting scheme. After identifying the principal direction, visual odometry can be obtained by using ICP to register two consecutive frames.

In section 3.2.1, more details for the principal direction detection are described. Section 3.2.2 shows how the detected principal direction is used to estimate the relative transformation between two consecutive frames. The current robot position in global coordinate can

3. MOTION ESTIMATION ON CEILING VISION

Figure 3.4: Lines extraction of ceiling image

be retrieved by simply accumulating the relative transformation matrix. In section 3.2.3, real world experiments are carried out to show the stability of proposed visual odometry estimation approach and its application to SLAM problem. Conclusions are drawn at last.

3.2.1 Principal direction detection

The principal direction detection refers to the detection of *yaw* (θ_z) value which is the rotation around the normal of the plane that the robot is currently on. The non-stationary objects, such as moving people, cannot be used to detect the principal direction of the image.

In order to have a better principal direction estimation, we mounted the Kinect on the robot and pointed it to the ceiling. Calibration and rectification are performed using the toolbox proposed in [104] to achieve undistorted frames.

Although the principal direction is a global feature of the environment, its calculation is based on local evidence extracted from monocular images. Edges have been recognized as critical features in image processing since the beginning of computer vision. Compared to point features, edges are less distinctive, which makes it more challenging for data association problem. However, the invariance of edges to orientation and scale makes them good candidates for tracking. Moreover, the runtime for edge detection is much faster compared to *scale-invariant feature transform* (SIFT) [99] or *speeded up robust features* (SURF) [100]. Canny’s algorithm [105] for choosing edgels in an image has emerged as the standard technique, and consistently ranks well in comparisons [106]. We use it for our edge feature selection algorithm. We first smooth the data using a symmetric 2D Gaussian blur kernel to remove some noise from the data (Figure 3.4(a)). After that, the Canny algorithm is adopted to generate an edge image (Figure 3.4(b)). We mounted a Kinect pointing up towards the ceiling. However, the data acquired may still contain parts from walls or lamps. These undesirable parts decrease the accuracy of principal direction detection. We eliminate this effect by extracting points only from the ceiling plane which cannot be correctly detected by using monocular images alone. As our data are grabbed from a Kinect, every point has a 2D position (pixel coordinates in the image) and its corresponding 3D position (the measured position in the world coordinate system). However, the ceiling plane cannot be determined through simple height filtering because the robot may have a non-zero roll (θ_x) or pitch (θ_y) angle. Our ceiling plane detection approach has two assumptions: First, the robot starts with $\theta_x = 0$ and $\theta_y = 0$, which means that at the start point, points from the ceiling can be determined through simple height filtering. Second, the slope

3. MOTION ESTIMATION ON CEILING VISION

(a) Smoothed monochrome image

(b) Canny edge detection

(c) Extracted edge pixels from ceiling

(d) Lines extracted by Hough Transform

(e) Rectified edge image

(f) Extracted lines overlapped on rectified edge image

Figure 3.5: Lines extraction under perspective distortion

change between two consecutive frames are small, which is reasonable if overall processing time for each frame is short. At the first frame, 3 non-collinear points at certain height are selected as points from the ceiling and a plane equation is estimated from these points. Then, for each following frame, the plane detection and outlier removal steps are listed as follows:

1. Use the plane calculated in the previous frame as initial hypothetical plane for the current plane estimation.
2. Points that are close enough to this an initial hypothetical plane are selected as set S .
3. A Ransac-fashioned approach [107] is applied to S to determine the current ceiling plane and eliminate all points that do not belong to this plane.
4. All the remaining 3D points are projected back to 2D image pixels. as shown in Figure 3.4(c).

Because the ceiling detection for the current frame is based on the last detected ceiling plane, this plane detection approach is adaptive to small changes in ceiling slope. If more than one planes (such as ceiling and walls) exists in the same frame, this approach can still recognize the ceiling correctly even when only a small portion of points are from the ceiling.

After projecting the 3D ceiling points back to the 2D image, the *Hough transform* is used to extract lines from these back-projected edge pixels. To reduce the effect of noisy data and to improve the algorithm stability, only those lines whose Hough votes are higher than a threshold are selected as shown in Figure 3.4(d). For the standard *Hough transform*, lines are expressed in polar system as:

$$r = x \cos \theta + y \sin \theta \quad (3.1)$$

where r is the distance from origin to the line and θ indicates the orientation of lines which are in the range $[0, 2\pi]$. Lines are detected

3. MOTION ESTIMATION ON CEILING VISION

with a voting of existence of edgelets in the image. As can be seen in Figure 3.4, edgelets are located along both horizontal and vertical directions. With this standard *Hough transform*, ambiguity is inevitable when determining the principal direction. Hence, a modified version of *Hough transform* is used here. We set angular resolution of θ_z as 1 degree. We made an assumption that the current detected principal direction is close to the previous detected one. This is a reasonable assumption since we use only two consecutive frames to calculate the transformation and the frame rate of Kinect camera can be up to 30 Hz. As a result, instead of searching the entire angular space for θ_z , the searching space is restricted to $[-\frac{\pi}{6}, \frac{\pi}{6}]$ relative to the principal direction in the previous frame, which results in much smaller searching space and faster line extraction process. Finally, the most voted angle value is selected as the principal directions.

However, if the robot has a non-zero *Roll* or *Pitch* angle, the perspective distortion will lead to a wrong estimation for the principal direction as shown in Figure 3.5(a) – 3.5(d). Lines that are parallel in the real world tend to intersect at some point. This error will lead to a wrong estimation of the principal direction. Fortunately, it can be corrected by utilizing the plane detected in the previous step.

Given detected ceiling plane $ax + by + cz + d = 0$, the plane normal can be expressed as

$$\mathbf{n} = \left(\frac{a}{\sqrt{a^2 + b^2 + c^2}}, \frac{b}{\sqrt{a^2 + b^2 + c^2}}, \frac{c}{\sqrt{a^2 + b^2 + c^2}} \right) \quad (3.2)$$

Roll (θ_x) and *Pitch* (θ_y) of the ceiling plane can be calculated using normal $\mathbf{n}$:

$$\theta_x = \arctan \frac{\mathbf{n}(1)}{\mathbf{n}(3)} \quad \theta_y = \arctan \frac{\mathbf{n}(2)}{\mathbf{n}(3)} \quad (3.3)$$

where $\mathbf{n}(i)$ denotes the i^{th} element in vector $\mathbf{n}$.

Given a 3D point in camera coordinate X , its corresponding pixel x in image is:

$$x = K [I|0] \begin{bmatrix} X \\ 1 \end{bmatrix} = KX \quad (3.4)$$

3.2 Visual odometry using principal direction

where K is the calibration matrix obtained in a previous calibration step. Rotation matrix R can be calculated by the estimated θ_x and θ_y as

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \quad (3.5)$$

Because the principal direction is the estimation for the robot's *yaw* rotation (θ_z), the rectification process needs to make the ceiling in the monocular image horizontal to the image plane ($\theta_x = \theta_y = 0$). The principal direction can be determined using the algorithm described before. The actual translation of camera does not affect the detection of principal direction. Thus, we assume there is no translation or θ_z changes between the distorted image and rectified image. The rectified pixel x' can be expressed as:

$$x' = K [R|0] \begin{bmatrix} X \\ 1 \end{bmatrix} = KRX \quad (3.6)$$

From equation (3.4) and (3.6) we get:

$$x' = KRK^{-1}x \quad (3.7)$$

where KRK^{-1} is a 3×3 homography matrix. The rectified edge image and its extracted lines can be seen in Figure 3.5(e) and 3.5(f) respectively.

The detected principal direction $\theta \in [0, 2\pi]$. Thus, one principal direction represents two possible headings of the robot. However, this orientation ambiguity does no harm to visual odometry estimation. Although the principal direction is used to guide a frame-to-frame based visual odometry estimation approach like ICP, the relative rotation changes between two consecutive frames cannot exceed π . Therefore, the ambiguity can be resolved while determining the relative transformation. θ_z is simply calculated as the relative change between the two consecutive principal directions.

3. MOTION ESTIMATION ON CEILING VISION

Figure 3.6: System overview.

3.2.2 Visual Odometry Implementation

Unlike feature matching based VO described in the previous section, data associations between two consecutive frames are not known. Given θ_x , θ_y and θ_z , we can roughly align two consecutive point clouds filled with edge points from the ceiling. After that, ICP is used to fine-tune the transformation parameters and provide 6-DOF (degree of freedom) motion estimation using these two pre-aligned point clouds. The system overview is shown in Figure 3.6.

In order to reach real-time performance, the size of point cloud needs to be reduced. We divided the space into a set of tiny 3D boxes and all points inside each box are approximated with the coordinates of their centroids. By default, we choose the voxel size to be of 0.05 m which allowed us to down-sample the point cloud by an order of magnitude, reaching real time requirement and having enough data for performing the alignment procedure later. Moreover, after this operation, points density no longer depends on their distances from

the sensor.

ICP has the drawback of local minima. Hence a reasonable initial guess of the transformation is needed in order to have a good estimation result. The detected principal direction θ_z together with θ_x and θ_y can be used as initial guess for rotation parameters. For translation, a simple alpha-beta filter based on the previous motion is applied. All points in the voxel grid are transformed into point cloud P_s based on this initial guess.

Given two pre-aligned point clouds, ICP is used to fine-tune the motion estimation. In ICP, points in a source cloud P_s are matched with their nearest neighboring points in a target cloud P_t and a rigid transformation is calculated by minimizing the n-D error between associated points. This transformation may change the nearest neighbors for points in P_s , so the two steps of association and optimization are alternated until convergence. P_s is transformed into P_s' using the pre-calculated initial guess. For each point in P_s' , data association is determined by finding the nearest neighbor in target cloud P_t . While it is possible to compute associations between points based on a combination of Euclidean distance, color difference, and shape difference, we found Euclidean distance along with a fast kd-tree search to be sufficient in most cases.

Point-to-plane ICP has been shown to generate more accurate alignments than point-to-point ICP due to an improved interpolation between points [108]. However, since almost all points in P_s' and P_t come from the ceiling, point-to-plane error metric is not feasible in our case and we employ the point-to-point error metric instead. The current robot location and orientation in global coordinate can be retrieved by accumulating the relative rotation and translation matrix.

3. MOTION ESTIMATION ON CEILING VISION

3.2.3 Experiments

In this section, several experiments are conducted to demonstrate the effectiveness and the stability of our principal direction detection approach for visual odometry estimation. The experiments were carried out using an iRobot Create platform. We used the Kinect camera which is pointed up to the ceiling to test our visual odometry system. The robot was manually controlled with a wireless game pad and the testing data were collected by driving the robot in the Robotic Research Center of Nanyang Technological University.

In each frame, around 8000 edge points were detected. Only about 3000 points were left after voxel filtering, and they were used to calculate the relative transformation between two consecutive frames. On a notebook with an Intel i7-2710-QE 2.10G Hz processor with 4GB of memory, the average number of frames per second for our visual odometry algorithm is about 10. Edge detection takes about 0.01 sec. It takes 0.009 sec and 0.007 sec for ceiling plane detection and principal direction detection respectively while 0.07 sec is required for ICP. Figure 3.7 shows the stability of our system under disturbance. During the test, the robot started at position (0,0) following a clockwise path and returned to the start point at the end. We manually disturbed the robot during the test. Figure 3.7(a) shows the *roll* and *pitch* angles at different times. In Figure 3.7(b), the straight-line path is the robot trajectory using the principal direction to guide ICP while the dotted one is without it. Although under obvious disturbance, the path of robot is still smooth and almost closed loop at the end. The bold parts of the paths are places where disturbance occurs. The test emphasizes the effectiveness of the principal direction detection.

We conducted more tests in common office scene to demonstrate the performance of our visual odometry system and made comparison with other conventional odometry options. The result is shown in Figure 3.8. The robot started at position (0,0), heading towards the

3.2 Visual odometry using principal direction

(a) Roll and pitch angle during robot's movement.

(b) Robot trajectory.

Figure 3.7: Robot trajectory with disturbance

3. MOTION ESTIMATION ON CEILING VISION

(a) Robot move straight forward.

(b) Robot move along "∞" shaped loop.

Figure 3.8: Comparison of proposed visual odometry approach and conventional odometry.

Figure 3.9: Comparison of visual odometry using feature-matching based approach and principal direction based approach.

positive X direction. Raw wheel odometry is the accumulated position changes over time calculated using only wheel encoders. Dead reckoning means to incorporate raw wheel odometry with IMU sensing. The visual odometry is the result of our proposed algorithm. In Figure 3.8(a), the path of raw wheel odometry and dead reckoning deviates a lot from the straight line because of the wheel slip and shaking of the robot while the visual odometry can successfully maintain on the straight line path regardless of the slip. In Figure 3.8(b), the robot started at $(0,0)$, heading towards the positive X direction. We drove the robot along a "∞" shaped loop and came back to the starting point in the end. Because errors are accumulated over time for both visual and wheel odometry, none of the odometry approaches managed to close the loop. However, visual odometry still outperforms conventional odometry methods.

Figure 3.9 shows that using proposed principal direction based approach, the accumulated error is reduced comparing to feature-matching

3. MOTION ESTIMATION ON CEILING VISION

Figure 3.10: Gmapping results when robot moves straight forward. From top to bottom are results based on raw wheel odometry, dead reckoning and visual odometry, respectively. Red circles indicate the robot’s starting points and blue circles indicate end points.

based approach. The robot started at position $(0, 0)$, heading towards the positive X direction. The ground truth was obtained by combining laser odometry and IMU sensing.

Odometry is commonly used in different SLAM algorithm. A more precise odometry can produce a better map. We fed the three odometry results mentioned above into the Gmapping algorithm [109]. A

3.2 Visual odometry using principal direction

Figure 3.11: Gmapping results when robot moves along "∞" shaped loop. From top to bottom are results based on raw wheel odometry, dead reckoning and visual odometry, respectively. Red circles indicate the robot's starting points and blue circles indicate end points.

3. MOTION ESTIMATION ON CEILING VISION

Hokuyo Laser Range Finder (Figure 3.1) is used to get measurements from surroundings and the produced map can be seen in Figure 3.10 and 3.11. The first row is the Gmapping result based on raw wheel odometry while the second and third rows are based on dead reckoning and visual odometry, respectively. Because of the wheel slip, the maps are distorted or have some false edges for raw wheel odometry and dead reckoning cases in varying degrees. Gmapping based on our visual odometry approach generates the best map. The reason is that, our visual odometry is more robust to disturbance due to the use of principal direction detection. In Figure 3.11, the bottom one shows that the loop successfully closed using Gmapping based on our visual odometry.

Although our approach can generate satisfactory results, the small errors still accumulate over time leading to the failure of loop closing. There are two main reasons for these failures. First, there are some long gaps in the logs (each lasting a second or more) when the logging disks are unable to sustain the frame rate. Not all of these gaps are successfully bridged by the algorithm, particularly when the image overlap is small. This is more of a system (hard disk) failure rather than a visual odometry algorithm failure. The second failure occurs when the robot enters a narrow corridor. In this case, only a small portion of the image is from the ceiling and the visual odometry works with few pixels or not enough overlap area with previous frames. These failures highlight a limitation of our approach. Visual odometry will only work in environments where enough ceiling overlap can be obtained between two consecutive frames.

3.3 Conclusion

In this chapter, a real-time method for determining the robot odometry only using ceiling vision is presented. Compared to conventional visual odometry approach that using feature matching between two

consecutive frames, it is clear that the introduction of principal direction detection has greatly improved the visual odometry performance and makes it robust and insensitive to camera disturbance. The method has been evaluated using the Kinect camera system. In an indoor environment with uneven floor, the precision of this visual odometry system is demonstrated to be better than wheel odometry and even better than the combination of wheel odometry and an IMU sensor, especially when wheel slip occurs. It also offers the prospect of substantially reducing the sensing cost. By using this approach we can replace very expensive inertial measurement units with a much cheaper Kinect camera and produce even better odometry results under many conditions. We also showed in experiments that the proposed visual odometry can significantly improve the SLAM performance.

After presented the capabilities and limitations of pure visual odometry, it should be noted that even the most minimal robot is likely to have some form of proprioceptive sensing (including encoders and/or IMU). Therefore, in future work, we intend to augment rather than replace these sensors, and to work with higher-level pose estimators that fuse data from multiple sources.

Chapter 4

Efficient Visual Odometry using Stereo Camera

In this chapter, we describe our efforts to improve the runtime performance of stereo visual odometry. As discussed in section 1.1, both monocular and stereo cameras can be used to estimate camera motion by finding feature correspondences from two consecutive images. However, for monocular approaches, the motion estimation can only be made up to an arbitrary scale factor. The fundamental matrix that describes the camera motion can be calculated using at least 7 or 8 feature correspondences. The camera rotation and translation can be computed through decomposition of the fundamental matrix. The rotation can be computed without ambiguity, however, it is impossible to compute translational scale using two consecutive images.

To deal with the ambiguity of scale, depth of the scene is required which is unavailable in unknown environments. However, the depth of the scene can be computed through triangulation using a stereo rig. Compared to monocular cameras, stereo cameras are able to triangulate the scene depth or 3D coordinates of detected feature points in one frame instead of two consecutive frames as in the case of monocular camera.

4. EFFICIENT VISUAL ODOMETRY USING STEREO CAMERA

Visual odometry can be used as a part of other more complicated applications, such as visual SLAM or 3D reconstruction. Hence, real-time performance is crucial for visual odometry estimation.

4.1 Feature Detection and Descriptor Extraction

4.1.1 Feature detection

The first step for visual odometry estimation is to detect features in captured images. Commonly, detectors use points with gradient peaks such as corners, however, edges, ridges, and contours are also used as keypoints. Each detector locates features with different degrees of invariance to attributes such as rotation, scale, perspective, occlusion, and illumination. There is no superior method for interest point detection for all applications. Sometimes corners are preferred over edges, since corner is a structure that can be used to compute an angular orientation for the feature.

A large number of corner feature detection approaches have existed in the literature. The Moravic corner detection algorithm is an early method of corner detection. A corner is found when there are changes in all directions, and an edge is found when all changes are found along the same direction. The correlation difference is calculated using the SSD between any two overlapping patches to calculate the amount of changes. This method is compute intensive.

The Harris [79] corner detector provides improvements over the Moravic method. The goal of the Harris method is to find the direction of fastest and lowest change for feature orientation, using a covariance matrix of local directional derivatives. However its is still computationally intensive for use in real-time applications of any complexity

4.1 Feature Detection and Descriptor Extraction

SIFT [99] and *SURF* [100] are very popular choices for visual feature detectors for their robustness against lighting and scale changes. However, without the help of a modern GPU, it is time consuming to detect SIFT or SURF features, which does not meet our purpose for fast visual odometry estimation. Moreover, if the processing speed for each frame is fast enough (up to 30Hz), the scale changes between two consecutive frames can almost be ignored. Hence, we use the *FAST* [95] feature detector with non-maximal suppression. In order to increase the processing speed and reduce the number of features that are very close to each other, the images are down-sampled before applying feature detection and using full-size images when performing motion estimation.

For each detected feature, *FAST* detector gives a response score based on its significance. We threshold the detected features into higher and lower response groups based on the response scores. This kind of grouping can be used to increase feature matching speed which will be described in section 4.2.

4.1.2 Descriptor extraction

In order to get rotation-invariant features, Sobel filter is often used to estimate feature orientations. It uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives along horizontal and vertical directions as shown in Figure 4.1.

In this chapter, a variation of Sobel filter with size 5×5 is used. If we define A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations respectively, the computations are as follows: For each frame, we first apply two 5×5 Sobel filters (Figure 4.2) across down-sampled image along horizontal and vertical directions. For each point in the

4. EFFICIENT VISUAL ODOMETRY USING STEREO CAMERA

Figure 4.1: Standard sobel kernel layout.

image, G_x and G_y are the resulting Sobel responses along horizontal and vertical directions. The feature direction can be obtained as $\text{atan2}(G_y, G_x)$ and the gradient magnitude can be represented as $G = \sqrt{G_x^2 + G_y^2}$. The descriptor is designed as concatenation of gradient magnitudes around detected features. Its layout consists of two circles with 16 coding positions as shown in Figure 4.3. The descriptor is encoded sequentially from inner circle to outer circle and both follow clockwise direction. The feature direction is quantized into 8 discrete directions which determine the starting coding position. The value for each dimension of the descriptor is scaled into $[0, 255]$ so that each descriptor can be represented by 16 bytes. By this simple design, the descriptor can be extracted efficiently and invariant to rotation and illumination changes.

4.2 Feature matching

The feature correspondences are found by comparing the similarity of corresponding descriptors. The similarity is defined as sum of absolute difference (SAD). In order to optimize the implementation, we use the *Streaming SIMD Extensions* (SSE) instructions which can perform

1	2	0	-2	-1	-1	-4	-6	-4	-1
4	8	0	-8	-4	-2	-8	-12	-8	-2
6	12	0	-12	-6	0	0	0	0	0
4	8	0	-8	-4	2	8	12	8	2
1	2	0	-2	-1	1	4	6	4	1

(a) Horizontal kernel. (b) Vertical kernel.

Figure 4.2: Smoothing and feature detection kernel layout.

Figure 4.3: Descriptor layout. Arrow in the middle indicates feature direction and number indicates coding sequence.

4. EFFICIENT VISUAL ODOMETRY USING STEREO CAMERA

16-byte operations very efficiently. Hence, the 16-byte SAD can be performed within one call of SSE instruction.

Although the descriptor similarity can be computed efficiently, finding the best matches for thousands of features is still time consuming. Several efforts are done to speed up matching process. In the feature detection step, the *FAST* detector can classify feature points into two classes based on whether the surrounding pixels are darker or brighter than the center pixel. The matchings are only be searched in the correct class. To further speed up the matching process, we divided the detected features into higher and lower response groups based on the *FAST* detection scores [95]. The number of features with lower detection score is much more than that of higher score. Moreover, the down-sampled images are divided into equally spaced 40×40 grids. In each grid, features with higher detection score are matched by finding minimum SAD between descriptors within the radius of 150 pixels. After that, a statistical result of maximum and minimum displacement of features in each grid can be obtained. With the statistical results, the matchings of the features with lower score can be searched efficiently in the searching area that bounded by the minimum and maximum feature displacement in the corresponding grid.

In order to further improve the matching accuracy and remove outliers, we adopt the circle matching approach proposed in [110]. As illustrated in Figure 4.4, features in the current left frame are first matched with the previous frame using feature matching proposed above. Next, matching is performed between previous left frame and previous right frame. Because the stereo camera is well calibrated, we can use epipolar constraints when perform the matching between left and right images. Similarly, matchings are found in the current right frame and finally back to current left frame. A matching is accepted only if the ending feature point is consistent with the starting feature point.

Figure 4.4: Circle matching of features in current and previous, left and right images.

In order to have better visual odometry estimation, image coordinates at full resolution is desired instead of that at half resolution. Hence, for each feature of circular matches, we double the image coordinates of 4 features. However, we observed that the feature coordinates at full resolution may be slightly off. To cope with this, we use feature descriptor distance to find best feature location within a 5×5 neighbourhood grid.

4.3 Motion estimation

Given the detected features and their correspondences, the general solution for motion estimation of stereo camera has been introduced in section 2.7 using either 3D-to-2D or 3D-to-3D correspondences. As stated in [7], motion estimated from 3D-to-2D correspondences is more accurate than that of 3D-to-3D correspondences. This is because the error of triangulation is only introduced once and thus the total error is reduced. The transformation T_k is estimated from 3D-to-3D correspondences X_{k-1} and p_k , where X_{k-1} the 3D coordinates of features detected in the $k-1$ th image I_{k-1} and p_k is image coordinates of features in image I_k . The formulation is to find T_k that minimizes

4. EFFICIENT VISUAL ODOMETRY USING STEREO CAMERA

the image reprojection error

$$\arg \min_{T_k} \sum_i \|p_k^i - \hat{p}_k^i\|^2 \quad (4.1)$$

where $\hat{p}_k^i$ is the reprojection of the 3D feature points X_{k-1}^i into image coordinates of I_k following rigid body transformation T_k of the following form:

$$T_k = \begin{pmatrix} R_k & t_k \\ 0 & 1 \end{pmatrix} \quad (4.2)$$

where $R_k \in \mathbf{SO}(3)$ is the rotation matrix, and $t_k \in \mathbb{R}^{3 \times 1}$ is the translation vector from X_{k-1} to X_k . Using perspective camera model and assuming no skew distortion, the reprojection $\hat{p}_k^i$ can be represented as

$$\begin{pmatrix} \hat{u}_k \\ \hat{v}_k \\ 1 \end{pmatrix} = K(T_k X_{k-1}) = \begin{pmatrix} f_x & 0 & c_u \\ 0 & f_y & c_v \\ 0 & 0 & 1 \end{pmatrix} \hat{X}_k \quad (4.3)$$

where f_x and f_y are the focal lengths, and c_u, c_v the image coordinates of the projection center. $\hat{X}_k$ is a 3D feature point in coordinate of frame $k - 1$ transformed to frame k according to T_k .

Because we cannot guarantee that the matchings are noise-free, we use RANSAC to eliminate the effects of outliers. As shown in [111], the minimal case to solve equation 4.1 involves three 3D-to-2D matches. Hence, we randomly choose three matches and estimate transformation T_k based on them using *LevenbergMarquardt* algorithm. All other matches whose reprojection errors are within certain threshold are chosen as inliers. This process repeats several iterations and winner is the one that generated most inliers. The final transformation T_k is computed again using all inliers from winning iteration.

4.4 Experiments and discussions

In this section, we present real-world experimental results of our visual odometry in the campus of Nanyang Technological University. A

Figure 4.5: The robot platform and stereo camera we used to conduct the experiments.

stereo camera with base width of approximately 12cm is mounted on a ground vehicle as shown in Figure 4.5.

The images are captured at 20 Hz frequency with size 640×480 . In each frame, around 1900 *FAST* features are detected. To increase the matching speed, about 400 detected features are selected using non-maximal suppression in a 3×3 window, to analyse the minimal and maximal displacement of features in corresponding grid as mentioned in section 4.2. After circle matching, around 400 feature matches are detected and only half of them are selected as inliers through *RANSAC* process as can be seen in Figure 4.6.

The computed visual odometry is shown in Figure 4.7. The robot was driven manually with a joystick and it followed a loop with approximately 100 meters. Although the ground truth is not available, we can still see that the trajectory of the robot almost closed at (0,0) in the end which proves the reliability of our visual odometry estimation approach. Two more experiments are done to compare our visual odometry results with GPS reading and results of FOVIS [14]

4. EFFICIENT VISUAL ODOMETRY USING STEREO CAMERA

Figure 4.6: Feature correspondences of previous left and current left frames. Green lines indicate inliers after *RANSAC* process and red lines are outliers.

as shown in Figure 4.8. FOVIS is an approach that follows a standard stereo visual odometry pipeline and integrates the techniques that provide a useful balance of speed and accuracy as detailed in Section 2.7.5. Although the GPS position is not very accurate as the GPS signals jump sometimes, it does not have the error accumulation problem as visual odometry does. Hence, for a long distance run, the GPS positions can be regarded as the ground truth, to validate the accuracy of visual odometry systems.

The experiments were carried out using one single core of i5 CPU at 3.2 GHz and the detail runtime for each step is listed in Table 4.1. The runtime for each step is averaged over 100 frames. Our visual odometry estimation approach is very efficient and can achieve over 25 Hz frequency. However, as can be seen in Table 4.1, the *RANSAC*-based motion estimation step is not as efficient as the other steps. This is because for *RANSAC* process, the number of iterations that is necessary to guarantee a correct solution can be computed as shown

(a) Robot moved along a clockwise loop.

(b) Robot moved along a "∞" shaped loop.

Figure 4.7: Visual odometry estimation result in the real-world experiment. The robot started at (0,0) and came back to starting point at the end.

4. EFFICIENT VISUAL ODOMETRY USING STEREO CAMERA

(a) Robot moved along a clockwise loop.

(b) Robot moved along a "∞" shaped loop.

Figure 4.8: Visual odometry comparison.

Step	Time
<i>FAST</i> feature detection	2.5 ms
<i>Sobel</i> filtering at half resolution	0.3 ms
<i>Sobel</i> filtering at full resolution	1.3 ms
Descriptor extraction	0.1 ms
Sparse feature matching	1.2 ms
Dense feature matching	6.1 ms
Project features to full resolution	2.2 ms
<i>RANSAC</i> -based motion estimation	24.9 ms
Total	38.6 ms

Table 4.1: Runtime analysis.

in [69]:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^s)}$$

where p is the probability of success and ϵ is the percentage of outliers in the data points and s is the total number of data points. As a result, many iterations are needed because of large number of incorrect feature correspondences.

4.5 Concluding Remarks

This chapter introduces an efficient visual odometry estimation approach. The visual odometry are estimated by accumulating the relative transformation between two consecutive frames. *FAST* features are detected and a rotation-invariant descriptor is extracted for each feature. One feature correspondence is accepted if a circular matching can be found. Visual odometry is estimated by minimizing the reprojection error between two consecutive frames and inliers are selected using *RANSAC*.

Experiments demonstrate the reliability and efficiency of our approach. However, it should be noticed that the *RANSAC*-based motion esti-

4. EFFICIENT VISUAL ODOMETRY USING STEREO CAMERA

mation is relatively inefficient compared to other step of the process.

Chapter 5

Self-calibration Approach for Wide Baseline Stereo Cameras using Sea Surface Images

In this chapter, we propose a self-calibration approach that can estimate camera rotation matrices for a stereo rig with wide baseline (3 meters) using the sea horizon and a 3D world point at infinite distance. Given images taken from left and right cameras, the relative roll and pitch angles between two cameras are recovered by aligning the sea horizon in the left and right images. The pitch angle is estimated by making the projections of one point at infinite distance appear at the same location in both images. A photometric minimization approach is applied to refine the rotation parameters. The intrinsic parameters of the stereo cameras can be obtained beforehand using the approach proposed in [112]. The horizon and one point at infinity are identified in both left and right images and the 3D and 2D geometric information of them is shown to be related to camera rotation parameters.

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

Conventionally, people use checkerboard to do camera calibration to find the intrinsic parameters, as introduced in [63]. The extrinsic parameters of a camera are determined after calibration by minimizing the reprojection error between image points and their corresponding 3D world coordinates as discussed in [65]. This kind of fixed calibration procedures give the most accurate results. However, in order to have a better calibration result, multiple images of a planar grid taken from different perspective is required if all the input points come from the same plane. It cannot be guaranteed that the calibration grid can cover the whole images in a pair of stereo images even though the calibration grid from different locations and perspectives are captured during the calibration process, especially in the case of wide baseline. As a result, the calibration will be off because the reprojection errors for those corners not appearing in the image are impossible to calculate. They also suffer from a number of other disadvantages. The procedure requires a special calibration object and user interaction. Problems may arise in other environments. For example, the parameters become useless after any changes in the camera positions due to e.g. mechanical vibration.

The intrinsic parameters represents the characteristics of the camera. Hence, it is independent of the scene. However, camera extrinsic parameters can be changed because of various reasons such as mechanical vibrations or temperature variations, and so on. As a result, even if we had enough equipments to operate such a complicated calibration process, we need to recalibrate the stereo rig every time we start the tracking system. Obviously, this would be extremely inefficient especially when the users of the system are not experts in computer vision. In order to have a reliable and long-term consistent system, a self-calibration approach which can automatically determine the extrinsic parameters is required. To cope with that, a method that allows recovery of the extrinsic parameters without using any special designed calibration pattern or objects is introduced.

Section 5.1 describes the estimation approach of relative rotation between left and right cameras. Section 5.2 discusses the effects of relative translation between two cameras on image rectification. Real-world experiments are demonstrated in section 5.3 to show the feasibility of proposed approach and the comparisons with checkerboard-based calibration approach.

5.1 Rotation parameters estimation

The approach that estimates the relative rotation parameters between two cameras using only the sea horizon and a point at infinite distance, is proposed in this section. The challenge of detecting sea horizons is that a horizon appears at a different location in every frame due to wave swell. The sea horizon is detected automatically using the technique proposed in [113], and the point correspondence at infinite distance is created with manual help in both left and right images. An overview of our rotation parameters estimation approach can be seen in Figure 5.1. The remainder of this section elaborates one each step.

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

Figure 5.1: Flowchart of proposed rotation estimation approach.

The stereo extrinsic parameters are composed of a rotation matrix $R \in SO(3)$ and a translation vector $t \in \mathbb{R}$. It defines the transformation of a 3D world point $X_l = (X, Y, Z)^T$ in the left camera coordinate to $X_r = RX_l + t$. The projections of this 3D world point in the left and right image are denoted as u_l and u_r . The distortion coefficients can be obtained in intrinsic parameter calibration process and for simplicity, the images that we referred to in the following are undistorted images. The relation of 3D scene points and projected 2D points on the image

is:

$$\begin{aligned} u_l &= K_l X_l \\ u_r &= K_r X_r \end{aligned} \tag{5.1}$$

where K_l and K_r are intrinsic parameters for left and right camera obtained beforehand. Stereo vision uses triangulation based on epipolar geometry to determine distance to an object. The depth can be estimated through disparity which is the position changes of an object when viewed from different cameras. In a normal stereo rig, finding correspondences of same object in two images requires a search in two-dimensions. However, if two cameras are aligned correctly to be coplanar, the search is simplified to one dimension - a horizontal line across the image and this camera alignment process is called stereo rectification. The rotations that make two camera coplanar applied to each camera are defined as R_l and R_r . The rectified 2D image points can be represented in homogeneous coordinate as:

$$\begin{aligned} \bar{p}_l &= K_l R_l X_l \\ \bar{p}_r &= K_r R_r X_r \end{aligned} \tag{5.2}$$

For computation simplicity, we apply transformation on left and right images such that both images have same intrinsic parameters K' :

$$\begin{aligned} \bar{p}_l' &= K' K_l^{-1} \bar{p}_l \\ \bar{p}_r' &= K' K_r^{-1} \bar{p}_r \end{aligned} \tag{5.3}$$

For the following, the pixel coordinates are all corrected coordinates as defined in equation (5.3).

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

Figure 5.2: The camera coordinate system.

The goal of our calibration process is to find R_l and R_r from a image pair such that the two cameras are coplanar and parallel to the sea horizon. After rectification, the sea horizon appears horizontal and in the middle of both images and a 3D world infinity point appears at the same location in both images. The camera coordinate system is shown in Figure 5.2. For each camera, the camera rotation angles with respect to X , Y and Z axes are α , θ and β respectively. The general rotation matrix R is generated by multiplying rotation matrices around each axes:

$$R = \begin{bmatrix} \cos\beta\cos\theta + \sin\alpha\sin\beta\sin\theta & -\cos\alpha\sin\beta & \sin\alpha\sin\beta\cos\theta - \cos\beta\sin\theta \\ \sin\beta\cos\theta - \sin\alpha\cos\beta\sin\theta & \cos\alpha\cos\beta & -\sin\alpha\cos\beta\cos\theta - \sin\beta\sin\theta \\ \cos\alpha\sin\theta & \sin\alpha & \cos\alpha\cos\theta \end{bmatrix} \quad (5.4)$$

For simplicity of representation we rewrite rotation matrix as:

$$R = \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix} \quad (5.5)$$

5.1.1 Derive θ (rotation around Y axis) from the horizon

Intuitively, if there is no rotation around the Y axis, the horizon should appear horizontally in the image. In other words, θ is closely related to the slope of the horizon in the image which is indeed true through our study. Let $p_1 = (u_1, v_1)$ and $p_2 = (u_2, v_2)$ be two image points on sea horizon in the left image. Its corresponding 3D point in the world coordinate on the sea horizon is $P_1^w = [x_1, y_1, 0]^T$ and $P_2^w = [x_2, y_2, 0]^T$ with y_1 and y_2 approaching infinity. Because only the relative rotation between two cameras is relevant, we can safely assume that the center of world coordinate system and the left camera coordinate system are the same. Rotation matrix that transforms from the world coordinate system to the left camera coordinate system is R_l . P_1^w is transformed from the world coordinate system to the left camera coordinate system as,

$$P_1^c = R_l^{-1} P_1^w = R_l^T P_1^w = \begin{bmatrix} Ax_1 + By_1 \\ Dx_1 + Ey_1 \\ Gx_1 + Hy_1 \end{bmatrix} \quad (5.6)$$

The projections of this 3D point P_1^w in left image is:

$$\begin{bmatrix} u'_1 \\ v'_1 \end{bmatrix} = \begin{bmatrix} f \frac{Ax_1 + By_1}{Dx_1 + Ey_1} + c_u \\ f \frac{Gx_1 + Hy_1}{Dx_1 + Ey_1} + c_v \end{bmatrix} \quad (5.7)$$

where f is the corrected focal length of left camera and (c_u, c_v) is the principal point. (u'_1, v'_1) is the corrected projection as in equation (5.3). Similarly, the projection of P_2^w in left image is:

$$\begin{bmatrix} u'_2 \\ v'_2 \end{bmatrix} = \begin{bmatrix} f \frac{Ax_2 + By_2}{Dx_2 + Ey_2} + c_u \\ f \frac{Gx_2 + Hy_2}{Dx_2 + Ey_2} + c_v \end{bmatrix} \quad (5.8)$$

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

Hence, in the left image, the slope of the sea horizon is:

$$\begin{aligned}\frac{v'_1 - v'_2}{u'_1 - u'_2} &= \frac{HD - GE}{BD - AE} \\ &= -\frac{\sin\theta_l}{\cos\theta_l} \\ &= -\tan\theta_l\end{aligned}\tag{5.9}$$

So, the slope of the sea horizon in the left image is just the tangent of the left camera's rotation around Y axis θ_l . Similarly, the right camera's rotation around Y axis θ_r can be obtained.

5.1.2 Derive α (rotation around X axis) from horizon

After obtaining θ_l and θ_r , we can rotate the left and right images accordingly such that $\theta_l = \theta_r = 0$ in the rotated images. So the rotation matrix of virtual camera that correspond to the rotated image becomes:

$$R' = \begin{bmatrix} \cos\beta & -\cos\alpha\sin\beta & \sin\alpha\sin\beta \\ \sin\beta & \cos\alpha\cos\beta & -\sin\alpha\cos\beta \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}\tag{5.10}$$

Since the image has been transformed such that $\theta = 0$, any point on the sea horizon $P_h^w = [x_h, y_h, 0]^T$ in the image appears on the same row v_h :

$$v_h = f \frac{Gx_h + Hy_h}{Dx_h + Ey_h} + c_v\tag{5.11}$$

If the camera aligned with world coordinate system, the sea horizon should appear close to the middle of the image. The rotation from world coordinate system to left and right camera coordinate system around X axis can be recovered from the difference between v_h and

row index of principal point c_v :

$$\begin{aligned} v_h - c_v &= f \frac{Gx_h + Hy_h}{Dx_h + Ey_h} \\ &= f \frac{G}{D} \\ &= -f \tan \alpha \end{aligned} \quad (5.12)$$

$$\Rightarrow \alpha = \arctan\left(\frac{c_v - v_h}{f}\right) \quad (5.13)$$

5.1.3 Derive β (rotation around Z axis) from point at infinity

Figure 5.3: Relation between infinity point and β .

After obtaining θ and α , we can rotate the image accordingly such that the θ and α in the rotated left and right images are zero. So there is one angle (β) left to be aligned as shown in Figure 5.3. Ω_1 is the current image plane with focal length f and Ω_0 is the desired rectified image plane with the same focal length. The projections of a 3D world point at infinity P on to Ω_0 and Ω_1 are u_0 and u_1 respectively. The searching for the point (at infinite distance) correspondence should along one horizontal line across two images in the

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

rotated images. However, the image appearance of the sea surface is similar, which makes the feature matching a challenging task. Even though the searching of this point correspondence is restricted to one dimension, the correct correspondence can not be guaranteed to be found. To further constrain the feature correspondence searching, a manual input is required. In the left image, the point that we want to find its correspondence in the right image, is selected manually. In the right image, the rough location (human is not good at precisely locate the feature location) of the same point is selected manually, as well. Given these manual inputs, the searching for the point correspondence is further constrained to 10 pixels around this manually selected location in the right image. In practice, it is found that, this manually selected feature correspondence requires minimal work but can guarantee a good calibration result.

If the two cameras are properly aligned u_0 and u_1 should appear at the same location in the left and right images. u_0 and u_1 are selected manually in both images. let $|u_0 - cu_0| = a$ and $|u_1 - cu_1| = b$, we have:

$$\tan(\gamma) = \frac{a}{f} \quad (5.14)$$

$$\tan(\beta + \gamma) = \frac{\tan\beta + \tan\gamma}{1 - \tan\beta\tan\gamma} = \frac{b}{f} \quad (5.15)$$

From equation (5.14) and (5.15) we can derive:

$$\tan(\beta) = \frac{f(b - a)}{f^2 + ab} \quad (5.16)$$

5.1.4 Rotation parameters refinement

After having the relative rotation parameters estimated from sea horizon and a point at infinity, we can further refine this rotation parameters based on the captured sea image pair. Given left and right images, I_l and I_r , we need the rigid body motion $g \in SE(3)$ that describes the relative camera motion between two images. The rigid

body motion should generate a mapping from pixels in I_l to pixels in I_r . Given the coordinate of a pixel in the left image $\mathbf{x}_l = (u_l, v_l)^T$, its corresponding 3D world point P_l with the reference of left camera can be represented as:

$$P_l = \Pi_l^{-1}(\mathbf{x}_l, d(\mathbf{x}_l)) \quad (5.17)$$

where $d(\mathbf{x}_l)$ is the corresponding depth of the pixel $\mathbf{x}_l$ and Π_l is the projection matrix of the left camera that projects a 3D point to 2D image coordinate in the form:

$$\Pi_l = \begin{bmatrix} f_u^l & 0 & c_u^l \\ 0 & f_v^l & c_v^l \\ 0 & 0 & 1 \end{bmatrix} \quad (5.18)$$

f_u^l and f_v^l are the focal lengths of the left camera. c_u^l and c_v^l are the principal points of the left image. Then equation (5.17) becomes:

$$P_l = d(\mathbf{x}_l) \left(\frac{u_l - c_u^l}{f_u^l}, \frac{v_l - c_v^l}{f_v^l}, 1 \right)^T \quad (5.19)$$

In the right camera coordinate system, the coordinate of P_l undergoes a rigid body motion that consists of a 3×3 rotation matrix $R \in SO(3)$ and a translation vector $t \in \mathbb{R}^3$. The coordinate of the same 3D point with the reference of the right camera is:

$$\begin{aligned} P_r &= RP_l + t \\ &= Rd(\mathbf{x}_l) \left(\frac{u_l - c_u^l}{f_u^l}, \frac{v_l - c_v^l}{f_v^l}, 1 \right)^T + t \end{aligned} \quad (5.20)$$

If we only consider very distant points which means $d(\mathbf{x}_l)$ is close to infinity, equation (5.20) can be simplified to:

$$\begin{aligned} P_r &= Rd(\mathbf{x}_l) \left(\frac{u_l - c_u^l}{f_u^l}, \frac{v_l - c_v^l}{f_v^l}, 1 \right)^T \\ &= Rd(\mathbf{x}_l) P_l^u \\ &= (d(\mathbf{x}_l) \mathbf{r}_1 P_l^u, d(\mathbf{x}_l) \mathbf{r}_2 P_l^u, d(\mathbf{x}_l) \mathbf{r}_3 P_l^u)^T \end{aligned} \quad (5.21)$$

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

where P_l^u represents the projection of P_l to unit depth. $\mathbf{r}_i$ denotes the i th row of R . In our case, the distant points can be considered points from the sky which correspond to the pixels above the sea horizon. The projection $\mathbf{x}_r$ of P_r onto the right image is:

$$\begin{aligned}\mathbf{x}_r &= \pi_r \frac{P_r}{d(\mathbf{x}_r)} \\ &= \pi_r \left(\frac{\mathbf{r}_1 P_l^u}{\mathbf{r}_3 P_l^u}, \frac{\mathbf{r}_2 P_l^u}{\mathbf{r}_3 P_l^u}, 1 \right)^T\end{aligned}\quad (5.22)$$

From equation (5.22) we can see that for image points with very distant depth in the left image, the depth of the pixel does not affect its projection on the right image.

To have a minimal parametrization of the relative rotation $R \in SO(3)$, we represent R using the vector $\xi = (\omega_1, \omega_2, \omega_3)$ which defines a twist:

$$\hat{\xi} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}\quad (5.23)$$

in the Lie algebra $se(3)$. The relative rotation can be obtained by the matrix exponential operation mapping from the Lie algebra to the Lie group as:

$$R = \exp(\hat{\xi})\quad (5.24)$$

Hence, the final warping function that transforms the pixel coordinate on the left image to right image is:

$$\begin{aligned}w(\xi, \mathbf{x}_l) &= \pi(R, P) \\ &= \pi_r(\exp(\hat{\xi}), \pi_l^{-1}(\mathbf{x}_l, d(\mathbf{x}_l)))\end{aligned}\quad (5.25)$$

Ideally, the warped left image should match exactly to the right image which means the matching pixels from left to right images should have the same intensity. However, due to various reasons such as noise and different white balance of two cameras, the match cannot be perfect. Hence, the relative rotation between two cameras can be estimated by minimizing the total intensity difference of the warped left image

and the right image. The residual of the i th pair of pixel matching is defined as:

$$r_i(\xi) = I_r(w(\xi, \mathbf{x}_l^i)) - I_l(\mathbf{x}_l^i) \quad (5.26)$$

where $I_r(\cdot)$ and $I_l(\cdot)$ are the intensity images grabbed from right and left cameras. Hence ξ can be obtained by minimizing the sum of the least squares function:

$$\begin{aligned} E(\xi) &= \sum r_i(\xi) \\ &= \sum_{\mathbf{x}_l^i \in \mathbf{X}_d} [I_r(w(\xi, \mathbf{x}_l^i)) - I_l(\mathbf{x}_l^i)]^2 \end{aligned} \quad (5.27)$$

where $\mathbf{X}_d$ denotes the pixels in the left image with very distant depth which can be considered as all pixels above the sea horizon. Since equation (5.27) is non-linear, we can use an iterative minimization procedure to build a first order approximation, by solving which the unknown rotation parameters can be estimated. The first order Taylor approximation of $r_i(\xi)$ is:

$$\begin{aligned} r_i^l(\xi) &= r_i(0) + \frac{\partial r_i(\xi)}{\partial \xi} \Big|_{\xi=0} \Delta \xi \\ &= r_i(0) + J_i \Delta \xi \end{aligned} \quad (5.28)$$

where J_i is the Jacobian of the i th pixel with respect to the rotation parameters which has dimension 1×3 . J_i can be decomposed as:

$$J_i = J_i^{I_r} J_i^p J_i^R \quad (5.29)$$

where $J_i^{I_r}$ is the image gradient of right image with respect to image coordinates which has dimension 1×2 . J_i^p is the derivative of perspective projection with respect to $3D$ point coordinate which has dimension 2×3 . J_i^R is the derivative of $3D$ point transformation with respect to rotation parameters which has dimension 3×3 . The minimum can be found by taking the derivative of equation (5.28) and setting it to zero, we can get:

$$\Delta \xi = (J_i^T J_i)^{-1} J_i^T r_i(0) \quad (5.30)$$

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

where $(J_i^T J_i)^{-1} J_i^T$ is the pseudo inverse of matrix J_i . Each pixel x_i can provide one equation in the form of equation (5.28). Given n such pixels and by stacking all the equations together we can get:

$$\Delta\xi = (J^T J)^{-1} J^T \mathbf{r}(0) \quad (5.31)$$

where $J \in \mathbb{R}^{n \times 3}$. In each iteration, the estimated rotation is updated by $\Delta\xi$ as:

$$\xi^{k+1} = \log(\exp(\xi^k) \exp(\Delta\xi)) \quad (5.32)$$

where ξ^k represents the estimated rotation in the k th iteration. Since the linearization in equation (5.28) is only valid for small motion ξ , we use a coarse-to-fine scheme: the rotation parameters are estimated on a coarse image scale initially and are refined on several finer scales iteratively. Hence even larger rotation can be handled.

5.2 Effects of translation vector

Given the estimated rotation matrix, the relative translation between two camera centers can be recovered using a calibration line segment and simple triangulation as discussed in [114]. In order to make left and right images to be coplanar in the rectified images, both left and right images need to rotate according to a rotation matrix R_{rec} such that the epipoles of both images goes to infinity and epipolar lines become horizontal. Given the translation vector $T = [t_x, t_y, t_z]$, R_{rec} is defined as:

$$R_{rec} = \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix} \quad (5.33)$$

where $e_1 = \frac{T}{\|T\|}$, $e_2 = \frac{1}{\sqrt{t_x^2 + t_z^2}}[-t_z, t_x, 0]$, $e_3 = e_1 \times e_2$. In many stereo systems, the left and right cameras are mounted in a way that they are almost parallel to each other. Especially in our stereo camera system with very wide baseline, which means t_x is very large (about 3 meters) compared to t_y and t_z (less than 0.05 meter). This allows us

to approximate t_y and t_z to zero which makes R_{rec} close to identity matrix.

The effects of t_y and t_z on depth estimation can also be ignored in our case. For example, $t_z = 0.05m$, given a point $P = [x, y, z]^T$ as shown in Figure 5.2, u_v of its projection in the image is $u_v = f \frac{z}{y}$. The projection of $P' = [x, y, z + t_z]$ in image is $u'_v = f \frac{z+t_z}{y}$. Given that the focal length of our cameras is 2800, the effect of t_z on pixel changes of a point with 300 meters depth can be represented as:

$$u_v - u'_v = f \frac{t_z}{y} \approx 0.47 \quad (5.34)$$

which can be safely ignored by allowing 1 pixel difference on rows when finding stereo matches along epipolar line. The effect of t_y on depth estimation is illustrated in Figure 5.4, O_1 and O_2 represent left and right cameras. Although they are parallel to each other, they are not perpendicular to the baseline B (3 meters) with a small angle φ . A perfect baseline should be B' . Given depth (Z) of a pixel, its disparity (d) is:

$$d = \frac{fB}{Z} \quad (5.35)$$

Given $t_x = 3.0$ meters and $t_y = 0.05$ meter we can calculate $\varphi \approx 1^\circ$. If the detected boat is located with a depth of 300 meters, the disparity difference of stereo rigs with baseline B and B' is:

$$\begin{aligned} d - d' &= \frac{f}{Z}(B - B') \\ &= \frac{f}{Z}B(1 - \cos(\varphi)) \end{aligned} \quad (5.36)$$

$$\approx 0.0043 \quad (5.37)$$

which is less than 1 pixel and cannot be observed in the images. t_x has no effects on rectification or stereo matching process. In our implementation, t_x is obtained by careful measurement. Although noise could exist in measurement, it only introduces small error to the final depth estimation. Moreover, This small error can be further reduced

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

Figure 5.4: Effects of translation on disparity

by using the distance from the camera to the target boat which can be measured using laser or GPS. Hence in our case, the effects of translation vector in the rectification process and depth estimation can be safely ignored.

5.3 Experimental results

Real world images are used to evaluate proposed approach. Four sequences of images with resolution 2736×2192 are captured for testing using a stereo rig mounted on Unmanned Surface Vehicle as shown in Figure 5.5. The sea horizon is detected automatically using a haar-like feature based approach ([113]) as shown in Figure 5.6. Any pair of captured image pair can be used for calibration. A sample image pair that we used for calibration is shown in Figure A.7(a). The pixel-wise intensity difference of images pairs before and after calibration process are depicted in Figure A.7(b) and A.7(d) respectively. As

Figure 5.5: Stereo rig mounted on an unmanned surface vehicle.

Figure 5.6: Horizon detection results.

introduced in section 5.1.4, the pixel-wise intensity difference should be minimized in the rectified images for pixels above sea horizon. In order to deal with larger rotation parameters, the optimization process is applied on 4 different image pyramids. The convergence rate and image scales are illustrated in Figure 5.8. The average intensity difference is computed using only pixels above sea horizon.

The rectification results of a sample image pair of our approach can be seen in Figure 5.9. If two images are properly rectified the epipolar lines should appear horizontal in rectified images. In order to evaluate such property, we compute the average intensity for each row and compare between original and transformed images as illustrated in

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

(a) Original right image overlaid on left image.

(b) Difference of original images.

(c) Rectified right image overlaid on left image.

(d) Difference of rectified images.

Figure 5.7: A sample calibration image pair.

Figure 5.8: Convergence rate of photometric error minimization process on multiple scales.

Figure 5.10. For the clarity of the figure, the row number is down-sampled to half of the full-size resolution and is indexed from top to bottom of the image. Given the estimated rotation parameters between two cameras, the left image can be transformed as seen from a position aligned with the right camera. The sea horizon appears at the 563th row in the rectified sample image pair. The intensity of rows above sea horizon from transformed left and original right images follows very similar distribution. The white balance difference of the two cameras introduces a small intensity value shift on the whole image.

In order to demonstrate the performance of the proposed approach, we compare it with the calibration approach using a checkerboard ($4m \times 3m$). Although it is very hard to carry out, we used a crane to rotate and move the checkerboard. The calibration using proposed approach was done on a sample image pair as shown in Figure 5.11(a)

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

(a) Before rectification (undistorted images).

(b) After rectification.

Figure 5.9: A sample rectified image pair with horizontal lines.

Figure 5.10: The average intensity for each row of down-sampled images.

and 5.11(b). After having rectified stereo images, the disparity images (Figure 5.11(c) and 5.11(d)) can be obtained using a stereo matching method such as SGBM [115]. Given disparity, the corresponding depth of image pixels can be estimated. Knowing the depth of a pixel p_l in the left camera, we can back-project it into 3D world coordinate P . Because the rotation between two cameras has been estimated, we can project P into right camera, hence, find p_r which is the correspondence of p_l in the right image as shown in Figure 5.11(e) and 5.11(f). Figure 5.12(a) and 5.12(b) show the projected left image overlaid on the original right image. The details of the overlaid images are illustrated in Figure 5.12(c) and 5.12(d) to demonstrate the performance of the calibration. With the same stereo matching approach, the proposed calibration result generate denser point cloud and better detail on the boat. We use the photometric difference between each p_l^i and p_r^i as the error metric to evaluate the performance of our calibration approach. The color-coded photometric difference are shown in

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

Figure 5.12(e) and 5.12(f) in which green to red represents intensity difference ranging from 0 to 10.

More images are captured to evaluate the proposed approach. We mounted the stereo rig on camera boat and use the stereo rig to track another target boat. Four image sequences of different scenarios are captured for evaluation: a) Camera boat followed target boat. b) Target boat moved away from camera boat to tanker ship. c) Target boat moved towards camera boat from sea shore. d) Target boat moved towards camera boat from tanker ship. 400 image pairs are captured for each sequence. The disparity density and the average of photometric difference are used to evaluate the calibration performance and the results are shown in Figure 5.13 and 5.14. Because the left and right images are taken from two cameras with different settings, even the pixels are perfectly matched, the photometric difference between them cannot be zero.

Although we use average photometric difference as a measurement to evaluate our approach, the actual ground truth is unknown. To cope with that, we manually create image sequences in a way that we transform the left images for each sequence to corresponding synthetic right images with a predefined rotation matrix. Given one left image and its synthetic right image, the rotation matrix is recovered using the proposed approach. We also use photometric difference as a measurement to evaluate our approach which should be zero if the rotation matrix is recovered perfectly. The tests are applied on the same 4 sequences and the results are shown in Figure 5.15.

After having the disparity image (Figure 5.11(d)), the actual distance for each pixel with valid disparity can be calculated through simple triangulation. The distance from the stereo rig to the target boat are calculated and compared with ground truth which is obtained through GPS devices mounted on both camera and target boats. Fig 5.16 shows the distance estimation results. From Figure 5.16(c), we can see that depth estimation is more accurate when the target boat is within

5.3 Experimental results

(a) Original left image.

(b) Original right image.

(c) Disparity image of left camera view (checkerboard).

(d) Disparity image of left camera view (proposed).

(e) Projection of left image to right camera view (checkerboard).

(f) Projection of left image to right camera view (proposed).

Figure 5.11: Comparison with checkerboard-based approach.

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

(a) Combination of projected left image and original right image (checkerboard).

(b) Combination of projected left image and original right image (proposed).

(c) Details of projected image (checkerboard).

(d) Details of projected image (proposed).

(e) Colour coded photometric difference of original right image and projected left image (checkerboard).

(f) Colour coded photometric difference of original right image and projected left image (proposed).

Figure 5.12: Comparison with checkerboard-based approach.

5.3 Experimental results

(a) Camera boat followed target boat.

(b) Target boat moved away from camera boat to tanker ship.

Figure 5.13: Intensity difference and stereo density of rectified sequences.

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

(a) Target boat moved towards camera boat from sea shore.

(b) Target boat moved towards camera boat with tanker ship as background.

Figure 5.14: Intensity difference and stereo density of rectified sequences.

5.3 Experimental results

(a) Camera boat followed target boat.

(b) Target boat moved away from camera boat to tanker ship.

(c) Target boat moved towards camera boat from sea shore.

(d) Target boat moved towards camera boat with tanker ship as background.

Figure 5.15: Intensity difference of synthetic image sequences.

5. SELF-CALIBRATION APPROACH FOR WIDE BASELINE STEREO CAMERAS USING SEA SURFACE IMAGES

300 meters. Moreover, we can observe that the depth estimation is step-shaped. The reason is that the disparity is discrete instead of continuous and the disparity is less sensitive to distant objects than close objects. Hence the step-shaped effect can be observed more clearly for distance above 300 meters.

5.4 Conclusion

In this paper we presented an approach to estimate the relative rotation for wide-baseline stereo cameras using a pair of stereo images. The advantages of the proposed approach are three fold: Firstly, only image pairs and no initial estimation of extrinsic parameters is required. Secondly, compared to approaches using a calibration pattern which requires heavy labor work to manipulate a huge calibration pattern and re-calibration is inevitable due to mechanical vibration etc., the proposed approach is very flexible and requires much less efforts and is easy to implement. Thirdly, although, the same point with infinite distance needs to be identified in both left and right images, the detection and matching of features are not necessary which makes the proposed approach especially suitable in featureless environment.

5.4 Conclusion

(a) Camera boat followed target boat with closer distance.

(b) Camera boat followed target boat.

(c) Target boat moved away from camera boat to tanker ship.

(d) Target boat moved towards camera boat with open sea as background.

(e) Target boat moved towards camera boat from sea shore.

(f) Target boat moved towards camera boat with tanker ship as background.

Figure 5.16: Distance estimation to target boat.

Chapter 6

Conclusions and Future work

This chapter describes the conclusions of this thesis. Further work is also discussed indicating various ways through which the research may continue.

6.1 Conclusions

This thesis focuses on the study of the stereo camera calibration and ego motion estimation for robotic applications. To this end, camera modelling and ways to calibrate stereo cameras as well as approaches to estimate homography and camera motion between two images are firstly reviewed.

A RGB-D camera is a popular choice to detect scene depth in a single frame. Many visual odometry methods based on RGB-D camera has been examined in the literature. However, no matter how accurate the motion estimation is, it still suffers from the drift problem, which means small errors of relative motion will accumulated overtime and result in false estimation of current camera's pose. Our efforts to reduce such accumulated error in an indoor environment are introduced.

6. CONCLUSIONS AND FUTURE WORK

Both of monocular and stereo cameras can be used to estimate camera motion by finding feature correspondences from two consecutive images. However, for monocular approaches, the motion estimation can only be made up to an arbitrary scale factor using at least 7 or 8 feature correspondences. The rotation can be computed without ambiguity, however, it is impossible to compute translational scale using two consecutive images. However, the depth of the scene can be computed through triangulation using a stereo rig. Compared to monocular cameras, stereo cameras are able to triangulate the scene depth or 3D coordinates of detected feature points in one frame instead of two consecutive frames as in the case of monocular camera. For that reason, our research focuses on stereo visual odometry and our efforts to improve the efficiency of stereo visual odometry approach are introduced.

The wider the stereo baseline, the higher the resolution of the disparity map would be which corresponds to more accurate depth estimation. However, from stereo calibration perspective, if conventional checkerboard based calibration approach is used, it cannot be guaranteed that the checkerboard can cover the whole images in a pair of stereo images for the case of wide baseline. To deal with that, a self-calibration approach which can automatically determine the extrinsic parameters for stereo camera with wide baseline is proposed.

Without knowing the camera parameters, two images of the same planar scene can be transformed from one to another using a homography which is estimated from feature correspondences between two images. False feature correspondences severely affect homography estimation result. RANSAC-based model fitting methods are designed to filter out outliers, however, they are time-consuming if a large portion of the data are outliers. an alternative robust and more efficient approach for homography estimation with the presence of outliers is introduced.

In this thesis, contributions have been made to the area of automatic self-calibration of stereo cameras and camera motion estimations. The main contributions and conclusions of this thesis are summarized as below:

- A real-time method for determining robot odometry only using ceiling vision is presented. A principal direction is detected from each ceiling image which can be used to reduce the drifting problem of visual odometry. Moreover, with the help of principal direction, our approach can perform well even with disturbance during robots movement. Experimental results have demonstrated its strength and this method can tolerate substantial disturbances in real run such as uneven floors. It offers the prospect of substantially reduced sensing costs. By using this approach we can replace very expensive inertial measurement units by much cheaper Kinect camera to produce even better odometry results for various purpose.
- An efficient visual odometry estimation approach is introduced. The visual odometry are estimated by accumulating the relative transformation between two consecutive frames. *FAST* features are detected and a carefully-designed descriptor is extracted for each feature which makes the descriptor to be rotation invariant and enable efficient descriptor comparison. One feature correspondence is accepted if a circular matching can be found. Visual odometry is estimated by minimizing the reprojection error between two consecutive frames and inliers are selected using RANSAC.
- An automatic approach for wide-baseline stereo extrinsic calibration using only a pair of sea surface images is introduced. Given intrinsic parameters and images taken from left and right cameras, by aligning sea horizon in left and right images the relative roll and pitch angles between two cameras can be recovered. The projections of one point with infinite depth are identified in both

6. CONCLUSIONS AND FUTURE WORK

images. The pitch angle is estimated by making the projections locate at the same pixel position in both images. Using the fact that large parts of images are projections of points from infinite distance, a photometric minimization approach is applied to refine the rotation parameters. The advantages of the proposed approach are three fold: Firstly, only image pairs and no initial estimation of extrinsic parameters is required. Secondly, compared to approaches using checkerboard which requires extra personnel to manipulate a huge checkerboard and re-calibration is inevitable due to mechanical vibration etc., the proposed approach is very flexible and requires much less efforts and easy to implement. Thirdly, feature correspondences are not necessary which makes the proposed approach especially suitable in featureless environment.

- The homography between image pairs are normally estimated by minimizing a suitable cost function given $2D$ keypoints correspondences. The correspondences are typically established using descriptor distance of keypoints. However, the correspondences are often incorrect due to ambiguous descriptors which can introduce errors into following homography computing step. There are numerous attempts to filter out these erroneous correspondences, but, it is unlikely to always achieve perfect matching. A novel homography estimation and outlier detection approach is proposed which is essentially different from conventional hypothesis-and-verify approaches such as RANSAC. We formulate the homography estimation and outlier detection problem together into a single nonlinear least squares problem. The new cost function combines both geometric error and descriptor similarity, by minimizing which, the homography and outliers can be determined simultaneously.

6.2 Future work

The work presented in this thesis proposes the following subjects for further research:

- In chapter 5, the sea horizon and a point at infinity are used to calibrate a stereo rig. Although the sea horizon can be detected automatically, the point at infinity is selected manually. In the future, it is desirable to design an automatic approach to detect the point at infinity. One difficulty in designing such approach is that, many points on the sea horizon have similar appearance. To make this problem easier, we can detect a distant boat near the sea horizon first, and then find point correspondences on this boat to estimate transformation between stereo cameras using RANSAC.
- Given a calibrated stereo rig on the sea surface, we are able to locate the locations of the target boats relative to our boat. However, the location of our boat is obtained from GPS, instead of visual inputs. In the future, we plan to apply stereo visual odometry on top of GPS localization to improve the localization accuracy.
- Although our visual odometry system can accurately estimate the motion of the robots, the small errors accumulate over time since there is no correction mechanism in the system. However, visual odometry can be considered as a necessary part of SLAM to correct the accumulated errors. The difference between visual odometry and visual SLAM is that: in visual odometry, only local consistency of the trajectory is concerned where as visual SLAM cares about the global map consistency. Hence, for the next step, we intended to build a visual SLAM system based on our visual odometry system.
- The reason why visual SLAM system can recover the accumulated errors is that when robot returns to a visited place (loop

6. CONCLUSIONS AND FUTURE WORK

closure), the difference between its estimated position and its true position can be identified and this information is used to reduce the drift in both the map and camera position. Hence, how to detect the loop closure in different lighting conditions is one of our future works.

- The Homography estimation approach in appendix A is achieved by minimizing a specially designed cost function. However, this approach has no special assumption of the transformation. Thus, the similar idea can be extended to other applications such as visual odometry estimation, which can be achieved by minimizing various cost functions as introduced in section 2.7.

Appendix A

Robust Homography Estimation based on Non-linear Least Squares Optimization

A.1 Introduction

Images in two views of world points lying on a planar surface are related by a homography matrix. Estimating a homography, given keypoints correspondences of image pairs, has received much attention due to its extensive applications. For example, in [46], the author used the homography estimated from planar objects, which are from images of various perspectives, to solve for the intrinsic and extrinsic parameters. It is also used in many other machine vision applications such as: panoramas generation [47], motion estimation [48], and augmented reality [49].

The homography between image pairs is normally estimated by minimizing a suitable cost function given 2D keypoints correspondences. The correspondences are typically established using descriptor dis-

A. ROBUST HOMOGRAPHY ESTIMATION BASED ON NON-LINEAR LEAST SQUARES OPTIMIZATION

tance of keypoints. However, the correspondences are often incorrect due to ambiguous descriptors which can introduce errors into the homography computation step. There have been numerous attempts to filter out these erroneous correspondences, but, it is unlikely to always achieve perfect matching.

Outlier removal approaches like RANSAC are intensively used to estimate better homography. However, RANSAC based model fitting methods are time-consuming if large number of outliers appears. Efforts can be made to reduce the number of outliers, however, such efforts require more computational power. Also, due to inevitable factors, such as repeated patterns, it is almost impossible to have outlier-free feature correspondences. Hence, an alternative robust and more efficient approach for homography estimation with the presence of outliers is desired. Here, we propose a non-linear least squares optimization approach to compute a homography, such that false matches have no or little effect on computed homography. Unlike normal homography computation algorithms as introduced in Section 2.5, our method formulates not only the keypoints' geometric relationship but also their descriptor similarity into the cost function. Moreover, the cost function is parametrized in such a way that incorrect correspondences can be simultaneously identified while the homography is computed. Experiments show that the proposed approach can perform well even with the presence of a large number of outliers.

The homography estimation given an image pair can be decomposed into two stages. In the first stage, keypoints in two images are detected and their local image descriptors are extracted. The keypoint matches are established by comparing the corresponding descriptors. After that, the false matches, due to ambiguous descriptors are detected and removed using robust methods. In the second stage, a cost function based on remaining matches is defined and the homography is computed by minimizing the cost function.

Much attention has been paid to the first stage. The descriptor ambiguity and reliability can be improved by using more distinctive descriptors such as PCA-SIFT [116]. However, feature matching approaches cannot completely resolve ambiguities especially in the case of repetitive patterns or occlusions. A hypothesize-and-verify framework such as Random sample consensus (RANSAC) [111] is the most popular method to remove inevitable outliers. It randomly and repeatedly selects and verifies a small set of matches to find inliers that are consistent with some global geometric constraints. Although during this process a homography can be estimated, this process is rather about outlier removal instead of the homography calculation. Usually all inliers are then fed into the optimization process again in the second stage to generate more accurate homography that can best explain the data.

More sophisticated variations have been proposed to improve the standard RANSAC algorithm. Some approaches such as PROSAC [117], ARRSAC [118] focus on improving the reliability of hypothesis. To achieve this, these methods use image appearance to select keypoint correspondences with high confidence and consequentially, speed up the search for consistent matches. Some other approaches such as MSAC [119], MLESAC [119] and MAPSAC [120] use more sophisticated measures such as mahalanobis distance instead of point consensus to better verify hypothesis. These approaches have succeeded in greatly reducing the error rate.

Given point correspondences, a cost function is formulated based on the difference between the measured and estimated image coordinates. A homography is obtained by minimizing this cost function using Direct Linear Transformation (DLT) or by iterative optimization methods such as Gauss-Newton or Levenberg-Marquardt. However, after the initial selection stage, it cannot be guaranteed that the data is noise-free before minimizing the cost function as there is no robust

A. ROBUST HOMOGRAPHY ESTIMATION BASED ON NON-LINEAR LEAST SQUARES OPTIMIZATION

rule to define outliers. Conventional least squares optimization algorithms are in general not robust to outliers. The reason is that Gaussian distribution of noise is the basic assumption for solving least squares problem and Gaussian distribution is sensitive to outliers due to its narrow tailed nature. Error increases quadratically in a least squares problem. Hence, the incorrect correspondences can easily lead to divergence of homography estimation. For example, a false point matching causes big geometric difference on the measured and estimated image coordinates. In order to compensate for this error, the homography has to deviate from its true value during the optimization process. Moreover, in some applications such as with ego motion estimation or circular panorama generation, such errors accumulate over time. Hence, even an extremely low error rate will eventually result in significant errors. As a result, given point correspondences, a robust cost function that can accurately calculate homography in the presence of outliers is needed. A desirable property for such an algorithm is that the incorrect correspondences have no or little contribution to the optimization process. In other words, during the homography calculation process, the algorithm should be able to identify the undetected mismatches as well.

Some robust cost functions such as Huber [121] have been proposed to reduce errors introduced by outliers. Unlike “normal” least squares in which the error has a quadratic influence on the cost function, the Huber function makes cost increase linearly if the error exceeds certain threshold. This means it weights large error less. However, the Huber cost function is not enough to deal with outliers, because the influence of outliers is reduced rather than removed.

Serradell et al.’s method [122] is able to solve correspondences and homography simultaneously. They combine a geometric prior and an appearance prior to achieve this. More specifically, the search space of the homography is constrained such as to limit the range of rotations and scales. Several homography hypothesis are sampled

in this search space. Gaussian Mixture Models that best fit these samples is formed as the geometric prior. The appearance prior is based on descriptor similarity distances. The homography and its covariance are iteratively updated by a Kalman filter that uses the best correspondences as measurements until the covariance becomes negligible. The potential mismatches are detected as those that are least likely to reduce, the covariances of the Kalman filter. In such a way, the influence of mismatches is removed, instead of reduced which makes it robust to high numbers of outliers. Their method can also be categorized into the “hypothesis-and-verify” approaches.

We propose a robust homography estimation approach that is different from the hypothesis-and-verify approach. It allows the homography calculation, and the incorrect correspondences identification simultaneously within a Bayesian framework. The contributions of the work presented here are:

- No initial samples or hypothesis of Homography are required.
- We formulate a new cost function which integrates keypoint consensus and descriptor similarity.
- We introduce a new set of parameters that represent the confidence for each correspondence, and by solving using non-linear optimization, both of the homography and false keypoint correspondences can be determined simultaneously.

With these improvements, our method can achieve robustness even when a rather high percentage of correspondences are incorrect. The influence of outliers for homography computing is much reduced, even removed, and thus can output a more accurate homography that satisfies most pixels’ geometric relationship between two images. Experimental results show the effectiveness and efficiency of the proposed approach in both synthetic images and real life images.

A.2 Cost Function Formulation

Given an image pair, $\mathbf{x}_i^j$ represents the i th point in j th image and its inhomogeneous image coordinate is denoted as (x_i^j, y_i^j) . The data set of extracted keypoint correspondences is $\mathbf{M}$. Each pair of corresponding points $\mathbf{x}_k^1$ and $\mathbf{x}_l^2$ defines a single point $\mathbf{m}_i$ in a measurement space $\mathfrak{R}^4$, formed by joining the coordinates in each image. The i th keypoint correspondence of image pair is represented by vector $\mathbf{m}_i = (x_k^1, y_k^1, x_l^2, y_l^2)^T$. The homography that transforms the first image to the second image is $\mathbf{H}$. According to Bayesian framework, $\mathbf{H}$, is estimated as:

$$Pr(\mathbf{H}|\mathbf{M}) = \frac{Pr(\mathbf{M}|\mathbf{H})Pr(\mathbf{H})}{Pr(\mathbf{M})} \quad (\text{A.1})$$

There is no prior knowledge about matchings and homography, we assume them to be uniformly distributed. Also, we assume that matchings are conditionally independent to each other. Hence, equation A.1 is changed into:

$$\begin{aligned} Pr(\mathbf{H}|\mathbf{M}) &\propto Pr(\mathbf{M}|\mathbf{H}) \\ &\propto \prod_{i=1 \dots n} Pr(\mathbf{m}_i|\mathbf{H}) \end{aligned} \quad (\text{A.2})$$

where n is the number of correspondences. Let $f(\mathbf{x}_k^1, \mathbf{H})$ be the estimation of point $\mathbf{x}_k^1$ projected on image 2 from image 1 by $\mathbf{H}$. The geometric error vector between the estimated and measured image coordinates introduced by matching $\mathbf{m}_i$ is defined as:

$$\mathbf{e}_i = \mathbf{x}_l^2 - f(\mathbf{x}_k^1, \mathbf{H}) \quad (\text{A.3})$$

Without loss of generality, the noise in the two images is assumed to be Gaussian on each image coordinate with zero mean and covariance matrix Σ . equation A.2 can be written as:

$$\begin{aligned}
Pr(\mathbf{M}|\mathbf{H}) &= \prod_{i=1:n} \left(\frac{1}{\sqrt{2\pi|\Sigma|}} \right) \exp\left(-\frac{1}{2}\mathbf{e}_i^T \Sigma^{-1} \mathbf{e}_i\right) \\
&= \left(\frac{1}{\sqrt{2\pi|\Sigma|}} \right)^n \prod_{i=1:n} \exp\left(-\frac{1}{2}\mathbf{e}_i^T \Sigma^{-1} \mathbf{e}_i\right) \quad (\text{A.4})
\end{aligned}$$

where n is the number of correspondences and $|\Sigma|$ is the determinant of covariance. Take the negative log likelihood of all the correspondences, equation A.4 becomes:

$$-\log Pr(\mathbf{M}|\mathbf{H}) = \eta + \frac{1}{2} \sum_{i=1:n} \mathbf{e}_i^T \Sigma^{-1} \mathbf{e}_i \quad (\text{A.5})$$

where η is a constant number. In order to optimally estimate $\mathbf{H}$, from equation A.1 to equation A.5, a MAP estimate $\mathbf{H}^*$ is made such that

$$\begin{aligned}
\mathbf{H}^* &= \arg \max_{\mathbf{H}} Pr(\mathbf{H}|\mathbf{M}) \\
&= \arg \min_{\mathbf{H}} -\log Pr(\mathbf{M}|\mathbf{H}) \\
&= \arg \min_{\mathbf{H}} \underbrace{\sum_{i=1:n} \mathbf{e}_i^T \Sigma^{-1} \mathbf{e}_i}_{\text{cost function } F(x)} \quad (\text{A.6})
\end{aligned}$$

where minimizing cost function $F(x)$ is a non-linear least squares problem and it can be solved using iterative methods such as Gauss-Newton or Levenberg-Marquardt. However, in this cost function, there is no mechanism to deal with outliers. Thus, false correspondences can easily cause the least squares optimization to converge to a wrong estimation.

It is desirable to reduce or ignore the influence of incorrect correspondences during optimization. Our main approach to achieve this is: as the optimization proceeds, we dynamically assign a corresponding reliability to each point correspondence. In other words, the contribution of each correspondence to the optimization problem changes while the

A. ROBUST HOMOGRAPHY ESTIMATION BASED ON NON-LINEAR LEAST SQUARES OPTIMIZATION

optimization proceeds. This leads to an extended optimization problem: not only parameters of homography need to be optimized but also the reliability of the feature correspondence.

Referring to cost function, Σ in equation A.6 represents the uncertainty of the measured image coordinates which is a good measure to control the portion of contribution of the corresponding point correspondence. For example, high uncertainty means the noise in the measured image coordinate is high and thus the error introduced by the corresponding keypoint match will contribute little to the optimization problem. However, in equation A.6 we can observe that Σ is constant for all correspondences. In order to make it change dynamically, we introduce a new set of parameters $\mathbf{C}$ to be optimized which describes the covariance matrix Σ_i for each measurement.

Moreover, the similarity of corresponding keypoints from two views is also included into cost function. The collection of similarity score is denoted as $\mathbf{S}$ and $\mathbf{s}_i$ is the similarity score of correspondence $\mathbf{m}_i$. The descriptor similarity is integrated into optimization problem in such a way that it helps to determine the values in $\mathbf{C}$ which describe the confidences of correspondences and $\mathbf{c}_i$ represents the confidence for match $\mathbf{m}_i$. With the idea, equation A.1 can be extended as:

$$\begin{aligned}
 Pr(\mathbf{H}|\mathbf{M}, \mathbf{C}, \mathbf{S}) &\propto Pr(\mathbf{M}, \mathbf{C}|\mathbf{H}, \mathbf{S}) \\
 &\propto Pr(\mathbf{M}|\mathbf{C}, \mathbf{H})Pr(\mathbf{C}|\mathbf{S}) \\
 &\propto \prod_{i=1 \dots n} \underbrace{Pr(\mathbf{m}_i|\mathbf{c}_i, \mathbf{H})}_{\text{likelihood}} \underbrace{Pr(\mathbf{c}_i|\mathbf{s}_i)}_{\text{prior}} \quad (\text{A.7})
 \end{aligned}$$

The first term is the likelihood of matching given the homography and uncertainty of measurements while the second term is the prior probability of $\mathbf{c}_i$ and the uncertainty of which is based on $\mathbf{s}_i$.

With the Gaussian assumption, the new cost function to be minimized

can be obtained by taking the negative log likelihood of equation A.7:

$$F'(x) = \sum_{i=1:n} (\underbrace{\mathbf{e}_i^T \Sigma_i^{-1} \mathbf{e}_i}_{\boldsymbol{\delta}_i} + \underbrace{\boldsymbol{\varepsilon}_i^T \Psi_i^{-1} \boldsymbol{\varepsilon}_i}_{\boldsymbol{\rho}_i}) \quad (\text{A.8})$$

with Ψ_i corresponding to covariance of prior distribution of $\mathbf{c}_i$. $\boldsymbol{\varepsilon}_i$ is the difference between the current $\mathbf{c}_i$ and its expected value $\bar{\mathbf{c}}_i$. We set $\bar{\mathbf{c}}_i$ to 1 for each correspondence.

Next, we elaborate on how $\mathbf{c}_i$ describes Σ_i . To simplify the problem, we assume that the error distribution of measurements along x and y directions are independent from each other and their standard deviations are the same. Thus $\mathbf{c}_i$ is defined as a scalar and it is enough to describe Σ_i . As a result, $\boldsymbol{\delta}_i$ in equation A.8 becomes:

$$\begin{aligned} \boldsymbol{\delta}_i &= \mathbf{e}_i^T \begin{bmatrix} \mathbf{c}_i^2 & 0 \\ 0 & \mathbf{c}_i^2 \end{bmatrix}^{-1} \mathbf{e}_i \\ &= (\mathbf{c}_i \cdot \mathbf{e}_i)^T \mathbf{I}_{2 \times 2} (\mathbf{c}_i \cdot \mathbf{e}_i) \end{aligned} \quad (\text{A.9})$$

$\mathbf{c}_i$ is defined as standard deviation of Σ_i . However, in equation A.9 we can see that it can also be interpreted as the weight of $\mathbf{e}_i$ which is the geometric error introduced by correspondence m_i . If m_i is an outlier, the corresponding $\mathbf{e}_i$ will contribute little to optimization problem by assigning a small weight to it and will be ignored if $\mathbf{c}_i = 0$, because in this case, $\mathbf{e}_i$ will not be added to the global error term. In this interpretation, if we scale $\mathbf{c}_i$ to $[0, 1]$, $\mathbf{c}_i$ not only makes optimization process robust to outliers, but also works as outlier and inlier selector as false $\mathbf{m}_i$ will make $\mathbf{c}_i$ close to 0 while correct $\mathbf{m}_i$ makes it close to 1.

Similarly as $\boldsymbol{\delta}_i$, we can write $\boldsymbol{\rho}_i$ in equation A.8 as:

$$\boldsymbol{\rho}_i = (\mathbf{s}_i \cdot \boldsymbol{\varepsilon}_i)^T \mathbf{I}_{2 \times 2} (\mathbf{s}_i \cdot \boldsymbol{\varepsilon}_i) \quad (\text{A.10})$$

The local feature descriptor pair of detected keypoints in $\mathbf{m}_i$ is $(\boldsymbol{\varphi}_k^1, \boldsymbol{\varphi}_l^2)$ in which $\boldsymbol{\varphi}_k^j$ is the descriptor vector of k th keypoint in image j . Let

A. ROBUST HOMOGRAPHY ESTIMATION BASED ON NON-LINEAR LEAST SQUARES OPTIMIZATION

$dist(\varphi_k^1, \varphi_l^2)$ represents the euclidean distance between descriptors φ_k^1 and φ_l^2 . $\mathbf{s}_i$ is defined as:

$$\mathbf{s}_i = \frac{\lambda}{1 + dist(\varphi_k^1, \varphi_l^2)} \cdot \frac{distNN_{\varphi_k^1}^{2nd}}{distNN_{\varphi_k^1}^{1st}} \quad (\text{A.11})$$

where $distNN^{1st}$ and $distNN^{2nd}$ denote euclidean distance between φ_k^1 and its first and second nearest neighbour in descriptor space. λ is a scale factor that we set to 0.03 manually. The ratio between first and second nearest neighbour can represent the distinctiveness of detected keypoints and thus is used to reduce the errors introduced repeated patterns.

Similar as $\mathbf{c}_i$, high $\mathbf{s}_i$ means more contribution of $\boldsymbol{\varepsilon}_i$ to optimization problem as is shown in equation A.10. The higher $\mathbf{s}_i$, the more difficult for $\boldsymbol{\varepsilon}_i$ to change which means the less likely that $\mathbf{c}_i$ deviates from its expected value $\bar{\mathbf{c}}_i$ which is 1. In other words, the higher the $\mathbf{s}_i$, the more likely that $\mathbf{m}_i$ is classified as a inlier.

The reason why we choose to work with descriptor similarity instead of appearance difference is two folds: first, the descriptors and distances to nearest neighbours have been computed before the optimization and thus saving computational time. Second, in this way, $\mathbf{S}$ will stay constant during optimization process, no matter how $\mathbf{H}$ and $\mathbf{C}$ change, because descriptors have been extracted beforehand. Hence, no more new parameters are needed to be optimized.

After introducing all variables of our cost function as in equation A.8, we can have a better intuitive understanding about it. $\boldsymbol{\delta}_i$ is responsible for minimizing the geometric error of homography transformation. $\mathbf{e}_i$ in it represents how inconsistent the correspondence $\mathbf{m}_i$ with estimated homography $\mathbf{H}$. In order to minimize it, high $\mathbf{e}_i$ which normally means outliers, will push $\mathbf{c}_i$ towards zero. Although it has the nice property of ignoring the errors introduced by outliers, even for low $\mathbf{e}_i$ which is non-zero, nothing prevents $\mathbf{c}_i$ reducing its value. This means that effects of all $\mathbf{m}_i$ are reduced even ignored during optimization

which will result in wrong estimation of homography. Fortunately, ρ_i helps to avoid this behaviour. It serves as a penalty whenever $\mathbf{c}_i$ decreases. This is achieved by setting all initial value of $\mathbf{c}_i$ and its expected value $\bar{\mathbf{c}}_i$ to 1. As a result, decreasing $\mathbf{c}_i$ corresponds to increasing prior error ε_i . Moreover, we introduce $\mathbf{s}_i$ to reflect similarity and distinctiveness of detected keypoints in $\mathbf{m}_i$ and the value of it determines the amount of penalty added to the cost function.

The least squares optimization can be further robustified. Due to Gaussian assumption, the error vector $\mathbf{e}_i$ and ε_i in equation A.8 have quadratic influence on cost function $F'(x)$. Even a single outlier would have major negative effect. In order to be more outlier robust, the Huber cost function is used and makes cost increase linearly if error exceeds certain threshold, thus equation A.8 becomes:

$$F'(x) = \begin{cases} F'(x), & \text{if } \sqrt{F'(x)} < b \\ 2b\sqrt{F'(x)} - b^2, & \text{otherwise} \end{cases} \quad (\text{A.12})$$

There are even more robust cost functions such as Blake-Zisserman, corrupted Gaussian or Cauchy. Compare to them, the reason why Huber function is preferred is that it does not introduce new local minima due to its convexity [69].

A.3 Sparsity of the system

The problem size of our approach increases with the number of key-point correspondences. Besides 9-parameter homography, for each new correspondence $\mathbf{m}_i$, a new parameter $\mathbf{c}_i$ that needs to be optimized is added to the problem formulation. Jacobian $\mathbf{J}$ and Hessian ($\mathbf{J}^T \mathbf{J}$) are required to minimize the cost function and their size also increases with the number of parameters. When solving $n \times n$ equations, it takes $O(n^3)$ for a dense system [123]. Thus, the size of the problem is not the bottle neck as solving it is much more crucial.

A. ROBUST HOMOGRAPHY ESTIMATION BASED ON NON-LINEAR LEAST SQUARES OPTIMIZATION

By exploiting the sparseness of the optimization problem, computational benefits can be gained through avoiding storing and operating on zero elements [124]. Hence, the sparser the structure, the more computational benefits we can gain.

Fortunately, when solving our minimization problems, the Jacobian and Hessian matrix in our problem have a sparse block structure as shown in Figure A.1. Only the black entries are non-zero. We can divide the Jacobian matrix into 4 zones. Top-left zone is the jacobian for geometric error $\mathbf{e}$ with respect to homography $\mathbf{H}$. The geometric error for each correspondence $\mathbf{e}$ is a 2D vector and the number of parameters of homography is 9. The top-right zone is the partial derivatives of $\mathbf{e}$ with respect to parameters $\mathbf{C}$ whose size equals to the number of correspondences. Because $\mathbf{c}_i$ only influences $\mathbf{e}_i$, the only non-zero entries in this area are $\frac{\partial \mathbf{e}_i}{\partial \mathbf{c}_i}$. The entries in bottom-left zone are partial derivatives of $\boldsymbol{\varepsilon}$ with respect to $\mathbf{H}$ and they are all zero. Because we only use descriptor similarity whose value is independent on homography estimation to prevent that $\mathbf{H}$ is involved into calculation of $\boldsymbol{\varepsilon}$. The bottom-right zone is diagonal and the only non-zero entries are the partial derivatives $\frac{\partial \boldsymbol{\varepsilon}_i}{\partial \mathbf{c}_i}$.

It is easy to calculate that the density of Jacobian matrix is $\frac{5}{9+n}$ while for Hessian it is $\frac{63+19n}{(9+n)^2}$. Hence the more the number of correspondences, the sparser the matrix would be which means more computational benefits can be obtained as we only operate on non-zero blocks.

A.4 Experiments and Evaluations

In this section we evaluate our approach using both synthetic and real-life pictures. Synthetics images are generated by transforming the original image I_s to target image I_t through a homography $\mathbf{H}$ generated randomly. We apply FAST [95] keypoint detector on I_s and I_t . BRIEF [125] descriptors are extracted from detected keypoints.

Figure A.1: Sparse structure example of Jacobian $\mathbf{J}$ (left) and Hessian $\mathbf{J}^T \mathbf{J}$ (right) of the proposed optimization problem. The black entries are non-zero. n is the number of keypoint correspondences. In this specific example, n equals to 10.

Keypoint correspondences $\mathbf{M}$ are determined simply by finding the nearest neighbours in descriptor space.

A.4.1 Synthetic noise

To evaluate the performance of our approach, besides the original outliers, more randomly generated false correspondences are added to $\mathbf{M}$. The homography estimation results of our approach are shown in Table A.1. The accuracy of estimated homography is evaluated using root mean-squared error (RMSE). From the results we can see that even the number of outliers is over ten times more than inliers, our homography estimation still remains feasible compared to the approach that minimizing the conventional cost function (RMSE w/o $\mathbf{C}$).

Our purpose to introduce $\mathbf{S}$ into cost function is to integrate descriptor similarity into optimization process and penalize the behaviour of decreasing $\mathbf{C}$. The lower the $\mathbf{s}_i$, the easier to decrease $\mathbf{c}_i$ and thus the easier $\mathbf{m}_i$ is classified as an outlier. The column “RMSE w/o $\mathbf{S}$ ”

A. ROBUST HOMOGRAPHY ESTIMATION BASED ON NON-LINEAR LEAST SQUARES OPTIMIZATION

Table A.1: Homography estimation results on random generated false matches. TP and FP mean true positive and false positive number while TN and FN represent true negative and false negative number. The column “RMSE w/o $\mathbf{S}$ ” represents the RMSE of approach without using descriptor similarity S while “RMSE w/o $\mathbf{C}$ ” represents the RMSE of standard optimization approach whose cost function is introduced in equation A.6.

True Matches	False Matches	TP	FP	TN	FN	RMSE	RMSE w/o $\mathbf{S}$	RMSE w/o $\mathbf{C}$
42	0	42	0	0	0	0.166	0.165	0.184
42	51	42	1	50	0	0.573	0.462	13.992
42	103	42	1	102	0	0.597	0.699	21.420
42	154	42	1	153	0	0.222	0.608	27.589
42	206	42	1	205	0	0.633	5.816	30.845
42	257	42	1	256	0	0.622	17.311	34.856
42	309	42	1	308	0	0.228	28.479	37.672
42	360	41	1	359	1	0.652	36.286	38.408
42	412	42	1	411	0	0.594	37.278	40.348
42	463	41	1	462	1	0.821	39.889	41.875
42	515	42	1	514	0	0.825	38.737	42.467

in Table A.1 shows that without $\mathbf{S}$, the algorithm still has some robustness against outliers even when the error rate is higher than 0.5, however, the RMSE increases significantly especially when the error rate goes higher. Although S is indispensable for the outlier selection in our approach, the selection is not totally depending on it. Figure A.2 shows all values of $\mathbf{s}_i$ in the case of 515 outliers. Even some outliers have high descriptor similarity and distinctiveness, they are still correctly classified as outliers after the algorithm converges. Please note that we limit the maximum value of $\mathbf{s}_i$ to 300 during the test.

When the optimization process finishes, a keypoint correspondence is classified as outlier if its $\mathbf{c}_i$ is below 0.5, since we initially set $\mathbf{c}_i$ to 1 and its value tends to move towards 0. False inliers deviate homography estimation more severely than false outliers because using a wrong correspondence during optimization process is much worse than not using a correct correspondence. The proposed approach can

Figure A.2: Value of s_i in the case of 515 outliers. The first 42 index are the ground truth inliers and the followings are ground truth outliers.

keep the number of false inliers very low compared to the total number of correspondences. Because many outliers are generated randomly, some of them are accidentally close to the truth as shown in Figure A.3. Such false matchings are acceptable to our algorithm. Because the error it introduced to the whole optimization problem is considerably small and any effort to reject such outliers would introduce new parameters to be tuned and thus complicate the problem.

It is worth noticing that more outliers does not necessarily means more error on homography estimation. For example, in Table A.1, the RMSE of the test with 309 outliers is lower than many of those with fewer outliers. The reason for this is that almost all of the c_i lies in the area that is either very close to 1 or very close to 0 as seen in Figure A.4(b). This means the effects of most outliers are removed instead of reduced while in the case of 206 outliers as shown in Figure A.4(a), c_i of some correspondences lies close to 0.5 which means the effect of some inliers is not enforced enough. Hence, even tests with fewer total outliers may still have higher RMSE.

The best case is to assign c_i of detected outliers to 0. However, this cannot be done naturally in our algorithm. In order to minimize

A. ROBUST HOMOGRAPHY ESTIMATION BASED ON NON-LINEAR LEAST SQUARES OPTIMIZATION

Figure A.3: Keyframe correspondences between original and transformed images with 567 outliers. The green lines indicate correctly detected inliers while blue means false negative and red means false positive detections. The correctly detected outliers are ignored for the sake of clearance of the figure.

Figure A.4: Distribution of $\mathbf{C}$ which represents the contribution of each matching to the optimization problem. Also, it is an indicator of outliers as we classify $\mathbf{m}_i$ as outlier if $\mathbf{c}_i < 0.5$. Without loss of generality, inliers are put in the first 42 index. (a) Test with 206 outliers in which 1 false inliers appeared. (b) Test with 309 outliers in which 1 false inliers appeared.

equation A.8, both $\mathbf{H}$ and $\mathbf{C}$ need to be optimized. As their Jacobians need to solve the optimization problem using an iterative method, the values of $\mathbf{C}$ during the optimization process have to be continuous instead of discrete. Hence, this inaccuracy is inevitable by just solving cost function. However, it is easy to refine the homography because our algorithm naturally assigns each matching a confidence $\mathbf{c}_i$ which is based on both the geometric error and descriptor similarity. We simply select the best m inliers to refine the homography using DLT, then the inaccuracy caused by continuous $\mathbf{C}$ can be removed.

We compare our approach with RANSAC and LMedS. Homography is estimated using the same data set as in Table A.1 and the result is shown in Figure A.5(a). As expected, the RMSE of LMedS rises when error rate is over 50%. With our robustified problem formulation, the RMSE of our approach keeps low (almost constant) with the increasing number of outliers. Our approach shows similar robustness compared to RANSAC and the same robustness can be easily achieved by using the refinement described above. We use optimized OpenCV

Figure A.5: Performance comparison with RANSAC and LMedS. (a) Root mean squared error comparison and (b) Runtime comparison.

implementations for both RANSAC and LMedS. We repeated tests 5 times for 11 noise levels and the run time is averaged for each level. The result is shown in Figure A.5(b). LMedS approach keeps low

A. ROBUST HOMOGRAPHY ESTIMATION BASED ON NON-LINEAR LEAST SQUARES OPTIMIZATION

and constant runtime at around 4 ms. The runtime for our approach started at 3.3 ms with 42 inliers and 0 outliers ended with 46.08 ms with 567 outliers. The RANSAC approach started at 0.81 ms and ended with 18249 ms. The runtime of RANSAC increases much faster than our approach because the required number of sampling times increases significantly with the portion of outliers [69].

The convergence property of proposed approach is depicted in Figure A.6. For homography estimation, compared to the standard cost function which reports geometric error only, our approach introduces additional parameters contributing to the overall error. As can be seen from the figure, our approach needs about 15 iterations to converge while the standard cost function for homography estimation needs significantly more iterations to be optimized. This indicates that compared to the standard cost function, our cost function has fewer local minimals and the newly introduced parameters offer faster convergence during optimization process.

Figure A.6: Number of iterations needed before optimization process converges. For our approach, the number of iterations is between 14 to 23.

A.5 Experiments on real life images

Table A.2: Experiment results on real pictures. The first pair of images are under different blur level and the second pair with different orientation and zoom. The third pair is for different view angles and repeated patterns.

Input	Reference	Our approach	RANSAC	Time(Ours)	Time(RANSAC)
16.79 ms	190.94 ms
13.08 ms	4525.1 ms
18 ms	57 ms

A.5 Experiments on real life images

Compared to synthetic images, Homography estimation is more complex for real life images due to illumination variation or camera distortions and so on. The robustness and efficiency of our approach are tested using real life images and the results are shown in Table A.2. The input and reference image pairs are under different viewing conditions. SIFT features are used to establish feature correspondences. Given the correspondences, homography between input and reference images is estimated using our approach and RANSAC. To evaluate the homography estimation, we transform the input image through the estimated homography and the differences between reference images and transformed images are shown in the third and fourth columns in Table A.2. We can see that the result is similar as the one in the synthesized experiment. Our approach can achieve similar robustness as RANSAC while saving much in computational time.

We also use Oxford VGG Graffiti image set for experiments as shown

A. ROBUST HOMOGRAPHY ESTIMATION BASED ON NON-LINEAR LEAST SQUARES OPTIMIZATION

Figure A.7: Oxford VGG Graffiti Images. Green dots are detected SIFT features.

in Figure A.7. SIFT features are used to generate feature correspondences for 3 image pairs: *image 1* to *image 2* (2679 matches), *image 2* to *image 3* (3061 matches), and *image 3* to *image 4* (3504 matches). Given a homography, keypoints can be projected into another plane. The accuracy of the homography estimation is measured by using the average Euclidean distance between the projected points using the true homography and the estimated homography as shown in Table A.3. The computation time of each homography estimation approach is shown in Table A.4. LMedS was also tested on these 3 image pairs, however, it failed in estimating homography from image pairs,

A.5 Experiments on real life images

since the inlier ratio is less than 0.5. RANSAC is the most accurate among these 3 estimators, however, it takes significantly longer time compared to the other two estimators. PROSAC ranks the correspondences by matching quality beforehand. It achieves large computational saving by drawing samples from top-ranked correspondences progressively, and keeps similar accuracy as RANSAC. However, for the pair “image3 to image4”, the efficiency gain for PROSAC is not as obvious as the other image pairs. The reason is that, PROSAC sorts data using the matching score. The descriptor similarity does not reflect matching quality properly in this case, because of the large view angle change. The performance of PROSAC approach in this case is similar to RANSAC as demonstrated in [117]. The proposed approach is the most efficient approach for the image pair “image3 to image4” and keeps similar accuracy as RANSAC and PROSAC. Because essentially, the proposed approach is not a hypothesis-and-verify approach compared to RANSAC and PROSAC. Also, it takes into account both the appearance similarity and geometric error during the optimization process as we discussed in Section A.2.

Table A.3: Accuracy on Oxford VGG Graffiti image set.

	image1 to image2	image2 to image3	image3 to image4
RANSAC	0.967	1.375	2.14
PROSAC	1.98	1.75	3.56
proposed approach	1.215	1.476	4.995

Table A.4: Computation time on Oxford VGG Graffiti image set.

	image1 to image2	image2 to image3	image3 to image4
RANSAC	648 ms	2092 ms	4432 ms
PROSAC	51 ms	262 ms	1284 ms
proposed approach	65 ms	309 ms	363 ms

A.6 Conclusion

We proposed a novel homography estimation and outlier detection approach which is essentially different from conventional hypothesis-and-verify approaches such as RANSAC. We formulate the homography estimation and outlier detection problem together into a single nonlinear least squares problem. The new cost function combines both geometric error and descriptor similarity, by minimizing which, the homography and outliers can be determined simultaneously.

Experiment results demonstrate that our approach achieves similar robustness as RANSAC under different viewing conditions. Due to the sparse structure of Jacobian and Hessian of proposed cost function, our algorithm remains efficient even with the presence of a large amount of outliers.

References

- [1] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus, “Interpreting and executing recipes with a cooking robot,” *Springer Tracts in Advanced Robotics*, vol. 88, pp. 481–495, 2013. 3
- [2] P. Fiorini and E. Prassler, “Cleaning and household robots: A technology survey,” *Autonomous Robots*, vol. 9, no. 3, pp. 227–235, 2000. 3
- [3] B. Hendy, “Hendy’s law,” 1998. [Online]. Available: <http://en.wikipedia.org/wiki/User:Barry.hendy/Sandbox> 4
- [4] E. Herbst, X. Ren, and D. Fox, “RGB-D flow: Dense 3-d motion estimation using color and depth,” in *IEEE International Conference on Robotics and Automation*, 2013, pp. 2276–2282. 4
- [5] S. Helmer and D. G. Lowe, “Using stereo for object recognition,” in *IEEE International Conference on Robotics and Automation*, 2010, pp. 3121–3127. 4
- [6] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Comput. Surv.*, vol. 38, no. 4, 2006. 4
- [7] D. Nistér, O. Naroditsky, and J. R. Bergen, “Visual odometry,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. 652–659. 5, 45, 46, 47, 50, 93

REFERENCES

- [8] M. Agrawal and K. Konolige, “Real-time localization in outdoor environments using stereo vision and inexpensive GPS,” in *18th International Conference on Pattern Recognition (ICPR)*, 2006, pp. 1063–1068. 5
- [9] I. P. Alonso, D. F. Llorca, M. Gaviln, S. A. Pardo, M. . G. Garrido, L. B. Vlacic, and M. . Sotelo, “Accurate global localization using visual odometry and digital maps on urban environments,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1535–1545, 2012. 5
- [10] N. Sünderhauf, K. Konolige, S. Lacroix, and P. Protzel, “Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle,” in *Autonome Mobile Systeme 2005, 19. Fachgespräch, Stuttgart, 8./9. Dezember 2005*, 2005, pp. 157–163. 5
- [11] K. Konolige and M. Agrawal, “Frameslam: From bundle adjustment to real-time visual mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066–1077. 5
- [12] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semi-direct monocular visual odometry,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 5
- [13] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli, “Depth mapping using projected patterns,” *Patent US20080240502*, 2012. 6
- [14] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Visual odometry and mapping for autonomous flight using an RGB-D camera,” *International Symposium on Robotics Research (ISRR)*, 2011. 7, 53, 95
- [15] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: Using depth cameras for dense 3d modeling of

- indoor environments,” in *International Symposium on Experimental Robotics (ISER)*, 2010, pp. 477–491. 8
- [16] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the RGB-D SLAM system,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1691–1696. 8
- [17] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, 1981, pp. 674–679. 8, 45
- [18] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234. 8
- [19] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in *IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136. 8
- [20] T. Whelan, H. Johannsson, M. Kaess, J. Leonard, and J. McDonald, “Robust real-time visual odometry for dense RGB-D mapping,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 5724–5731. 8
- [21] F. Steinbrucker, J. Sturm, and D. Cremers, “Real-time visual odometry from dense RGB-D images,” in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 719–722. 8
- [22] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for RGB-D cameras,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3748–3754. 8

REFERENCES

- [23] C. Audras, A. Comport, M. Meilland, and P. Rives, “Real-time dense RGB-D localisation and mapping,” in *Australian Conference on Robotics and Automation (ACRA)*, 2011. 8
- [24] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580. 8
- [25] H. Strasdat, J. Montiel, and A. J. Davison, “Scale drift-aware large scale monocular slam.” in *Robotics: Science and Systems*, vol. 2, no. 3, 2010, p. 5. 8
- [26] K. Konolige and M. Agrawal, “Frameslam: From bundle adjustment to real-time visual mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066–1077, 2008. 8
- [27] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, “Visual simultaneous localization and mapping: a survey,” *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, 2015. 8
- [28] H. Zhang, B. Li, and D. Yang, “Keyframe detection for appearance-based visual slam,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2071–2076. 9
- [29] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, “A comparison of loop closing techniques in monocular slam,” *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1188–1197, 2009. 9
- [30] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, 1st ed. MIT Press, 1993. 10, 29

- [31] P. F. Sturm and S. J. Maybank, “On plane-based camera calibration: A general algorithm, singularities, applications,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 1999, p. 432437. 10
- [32] J.-Y. Bouguet, “Camera calibration toolbox for matlab,” in http://www.vision.caltech.edu/bouguetj/calib_doc/. 10
- [33] T. Marita, F. Oniga, S. Nedevschi, T. Graf, and R. Schmidt, “Camera calibration method for far range stereovision sensors used in vehicles,” in *Intelligent Vehicles Symposium*, 2006, pp. 356–363. 10
- [34] Q. T. Luong and O. D. Faugeras, “Self-calibration of a moving camera from point correspondences and fundamental matrices,” *International Journal of Computer Vision*, vol. 22, pp. 261–289, 1997. 12
- [35] R. Hartley, “Self-calibration from multiple views with a rotating camera,” *European Conference on Computer Vision*, pp. 471–478, 1994. 12
- [36] L. Ran, Z. Hua, L. Manlu, X. Xianfenga, and H. Tianlian, “Stereo cameras self-calibration based on sift,” in *IEEE International Conference on Measuring Technology and Mechatronics Automation*, 2009, pp. 352–355. 12
- [37] B. Mårten and J.-O. Eklundh, “Real-time epipolar geometry estimation of binocular stereo heads,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 425–432, 2002. 12
- [38] T. Dang, C. Hoffmann, and C. Stiller, “Continuous stereo self-calibration by camera parameter tracking,” *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1536–1549, 2009. 12

REFERENCES

- [39] Y. Liu, T. S. Huang, and O. D. Faugeras, “Determination of camera location from 2-d to 3-d line and point correspondences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 13–27, 1990. 12, 28
- [40] H. Zhuang and Y. Shiu, “A noise-insensitive algorithm for robotic hand/eye calibration with or without sensor orientation measurement,” *IEEE Trans. Systems, Man and Cybernetics*, vol. 23, pp. 1168–1174, 1993. 12
- [41] H. Chen, “Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 530–541, 1991. 12
- [42] S. Li and Y. Hai, “Easy calibration of a blind-spot-free fisheye camera system using a scene of a parking space,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 232–242, 2011. 12
- [43] J. M. Collado, C. Hilario, A. de la Escalera, and J. M. Armingol, “Self-calibration of an on-board stereo-vision system for driver assistance systems,” in *Intelligent Vehicles Symposium*, 2006, pp. 156–162. 12
- [44] B. Musleh, D. Martin, J. Armingol, and A. de la Escalera, “Extrinsic parameter self-calibration and nonlinear filtering for in-vehicle stereo vision systems at urban environments,” in *International Conference on Computer Vision Theory and Applications*, 2014, pp. 427–434. 12
- [45] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981. 13

- [46] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000. 137
- [47] M. Brown and D. G. Lowe, “Automatic panoramic image stitching using invariant features,” *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007. 137
- [48] D. Nister, “Preemptive ransac for live structure and motion estimation,” *International Conference on Computer Vision*, pp. 199–206, 2003. 137
- [49] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, “Pose tracking from natural features on mobile phones,” *International Symposium on Mixed and Augmented Reality*, pp. 125–134, 2008. 137
- [50] M. Ito, “Robot vision modelling-camera modelling and camera calibration,” *Advanced Robotics*, vol. 5, no. 3, pp. 321–325, 1991. 17
- [51] E. L. Hall, J. B. K. Tio, C. A. McPherson, and F. A. Sadjadi, “Measuring curved surfaces for robot vision,” *Computer Journal*, vol. 15, no. 12, pp. 42–54, 1982. 17
- [52] R. Szeliski, *Computer vision: algorithms and applications*, 1st ed. Springer, 2010. 26
- [53] A. Fusiello, E. Trucco, and A. Verri, “A compact algorithm for rectification of stereo pairs,” *Machine Vision and Applications*, vol. 12, pp. 16–22, 2000. 27
- [54] R. K. Lenz and R. Y. Tsai, “Techniques for calibration of the scale factor and image center for high accuracy 3d machine vision metrology,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 713–720, 1988. 28

REFERENCES

- [55] M. Penna, “Camera calibration: A quick and easy way to detection the scale factor,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 12, pp. 1240–1245, 1991. 28
- [56] C. C.Wang, “Extrinsic calibration of a vision sensor mounted on a robot,” *IEEE Int. Journal on Robotics and Automation*, vol. 8, no. 2, pp. 161–175, 1992. 28
- [57] R. G. R. Hartley and T. Chang, “Stereo from uncalibrated cameras,” in *Computer Vision and Pattern Recognition*, 1992, pp. 761–764. 28
- [58] R. I. Hartley, “Estimation of relative camera position for uncalibrated cameras,” in *European Conference on Computer Vision*, 1992, pp. 579–587. 28
- [59] Q.-T. Luong and T. Vieville, “Canonic representation for geometries of multiple projective views,” in *European Conference on Computer Vision*, 1994, pp. 589–599,. 28
- [60] Z. Zhang, “Determining the epipolar geometry and its uncertainty: A review,” *International Journal of Computer Vision*, vol. 27, no. 2, pp. 161–198, 1998. 28
- [61] P. H. S. Torr and D. W. Murray, “The developement and comparison of robust methods for estimating the fundamental matrix,” *International Journal of Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997. 28
- [62] R. Y. Tsai, “A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses,” *IEEE Journal of Robotics and Automation*, vol. 3, pp. 323–344, 1987. 29

- [63] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, 2000. 29, 102
- [64] K. Mhlmann, “Design und implementierung eines systems zur schnellen rekonstruktion dreidimensionaler modelle aus stereo-bildern,” *PhD thesis, University of Mannheim*, 2002. 33
- [65] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*, 2nd ed. Cambridge University Press, 2004. 33, 35, 36, 39, 43, 102
- [66] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2011. 35
- [67] Q.-T. Luong and O. D. Faugeras, “The fundamental matrix: Theory, algorithms, and stability analysis,” *International Journal of Computer Vision*, vol. 17, no. 1, pp. 43–75, 1996. 35
- [68] R. Hartley, “Theory and practice of projective rectification,” *International Journal of Computer Vision*, vol. 35, p. 115127, 1999. 35
- [69] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004. 39, 51, 66, 99, 147, 154
- [70] J. J. Lee and G. Y. Kim, “Robust estimation of camera homography using fuzzy ransac,” *International Conference on Computational Science and Its Applications*, 2007. 42
- [71] P. J. Rousseeuw, “Least median of squares regression,” *Journal of the American Statistical Association*, vol. 79, pp. 871–880, 1984. 43

REFERENCES

- [72] S. Lacroix, A. Mallet, R. Chatila, and L. Gallo, “Rover self localization in planetary-like environments,” in *Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 1999, pp. 433–440. 43, 46
- [73] L. Matthies and S. Shafer, “Error modeling in stereo navigation,” *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, pp. 239–250, June 1987. 43
- [74] L. H. Matthies, “Dynamic stereo vision,” Ph.D. dissertation, Pittsburgh, PA, USA, 1989, aAI9023429. 43
- [75] H. Moravec, “Obstacle avoidance and navigation in the real world by a seeing robot rover,” in *tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University, doctoral dissertation, Stanford University*, September 1980, no. CMU-RI-TR-80-03. 43
- [76] C. F. Olson, L. Matthies, M. Schoppers, and M. W. Maimone, “Robust stereo ego-motion for long distance navigation,” in *Proceedings of Computer Vision and Pattern Recognition*, 2000, pp. 2453–2458. 43, 46
- [77] H. P. Morevec, “Towards automatic visual obstacle avoidance,” in *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, San Francisco, CA, USA, 1977, pp. 584–584. 44
- [78] W. Förstner, “A feature based correspondence algorithm for image matching,” in *Int. Arch. of Photogrammetry and Remote Sensing*, vol. 26, no. 3, 1986, pp. 150–166. 44
- [79] C. G. Harris and J. M. Pike, “3d positional integration from image sequences,” in *Proceedings of the Alvey Vision Conference*, 1988, pp. 87–90. 44, 62, 88

- [80] T. Suzuki and T. Kanade, “Measurement of vehicle motion and orientation using optical flow,” in *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, 1999, pp. 25–30. 45
- [81] F. Woelk and R. Koch, “Robust monocular detection of independent motion by a moving observer,” in *Proceedings of the 26th DAGM Symposium*, 2004, pp. 27–35. 45
- [82] G. P. Stein, O. Mano, and A. Shashua, “A robust method for computing vehicle ego-motion,” in *IEEE Intelligent Vehicles Symposium*, 2000, pp. 362–368. 45
- [83] R. Goecke, A. Asthana, N. Pettersson, and L. Petersson, “Visual vehicle egomotion estimation using the Fourier-Mellin transform,” in *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 450–455. 45
- [84] M. J. Milford and G. F. Wyeth, “Single camera vision-only slam on a suburban road network,” in *IEEE International Conference on In Robotics and Automation*, 2008, pp. 3684–3689. 45
- [85] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE Conference on Computer Vision and Pattern Recognition.*, pp. 195–202, 2003. 45
- [86] P. I. Corke, D. Strelow, and S. Singh, “Omnidirectional visual odometry for a planetary rover,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004. 45
- [87] M. Lhuillier, “Automatic structure and motion using a catadioptric camera,” in *IEEE Workshop on Omnidirectional Vision*, 2005, pp. 1–8. 45

REFERENCES

- [88] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, “Real time localization and 3d reconstruction,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 363–370. 45
- [89] L. Matthies and S. Shafer, “Error modeling in stereo navigation,” *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, pp. 239 – 250, June 1987. 46
- [90] Y. Cheng, M. Maimone, and L. Matthies, “Visual odometry on the mars exploration rovers,” *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 54–62, 2006. 46
- [91] M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the mars exploration rovers: Field reports,” *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007. 46
- [92] A. Milella and R. Siegwart, “Stereo-based ego-motion estimation using pixel tracking and iterative closest point,” in *Proceedings of IEEE International Conference on Computer Vision Systems*, 2006, pp. 21–24. 46
- [93] J. Shi and C. Tomasi, “Good features to track,” in *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR’94)*, 1994, pp. 593–600. 46
- [94] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992. 46
- [95] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European Conference on Computer Vision*, vol. 1, May 2006, pp. 430–443. 53, 89, 92, 148
- [96] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, “A constant time efficient stereo slam system,” in *British Machine Vision Conference*, 2009. 53

- [97] A. Howard, “Real-time stereo visual odometry for autonomous ground vehicles,” in *International Conference on Intelligent Robots and Systems*, 2008, pp. 3946–3952. 54
- [98] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the RGB-D slam system,” in *IEEE International Conference on Robotics and Automation*, 2012. 62
- [99] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *international Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. 62, 71, 89
- [100] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded up robust features,” in *European Conference on Computer Vision*, 2006, pp. 404–417. 62, 71, 89
- [101] N. Burrus, “Kinect RGBDemo v0.6.1,” 2012. [Online]. Available: <http://burrus.name/index.php/Research/KinectRgbDemoV6> 63
- [102] D. Kalman, “A Singularly Valuable Decomposition: The SVD of a Matrix,” *The College Mathematics Journal*, no. 1, pp. 2–23, 1996. 65
- [103] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments,” *International Journal of Robotics Research (IJRR)*, vol. 31, no. 5, pp. 647–663, 2012. 68
- [104] D. Herrera C, J. Kannala, and J. Heikkila, “Joint depth and color camera calibration with distortion correction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 2058–2064, 2012. 71

REFERENCES

- [105] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986. 71
- [106] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, “Comparison of edge detectors: A methodology and initial study,” in *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, 1996, pp. 143–148. 71
- [107] S. Se and M. Brady, “Ground plane estimation, error analysis and applications,” *Robotics and Autonomous Systems*, vol. 39, no. 2, pp. 59–71, 2002. 73
- [108] A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp,” in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009. 77
- [109] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with Rao-Blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007. 82
- [110] A. Geiger, J. Ziegler, and C. Stiller, “StereoScan: Dense 3D reconstruction in real-time,” in *Intelligent Vehicles Symposium (IV)*, 2011. 92
- [111] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. 94, 139
- [112] J. Heikkilä and O. Silvén, “A four-step camera calibration procedure with implicit image correction,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112, 1997. 101

- [113] Y. Yan, B.-S. Shin, X. Mou, W. Mou, and H. Wang, "Efficient horizon detection on complex sea for sea surveillance," *International Journal of Electrical, Electronics And Data Communication*, vol. 3, no. 12, pp. 49–52, 2015. 103, 116
- [114] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *International journal of computer vision*, vol. 4, no. 2, pp. 127–139, 1990. 114
- [115] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 328–341, 2008. 121
- [116] Y. Ke and R. Sukthankar, "PCA-sift: A more distinctive representation for local image descriptors," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 506–513, 2004. 139
- [117] O. Chum and J. Matas, "Matching with prosac - progressive sample consensus," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 220–226, 2005. 139, 157
- [118] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus," *European Conference on Computer Vision*, pp. 500–513, 2008. 139
- [119] P. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, pp. 138–156, 2000. 139
- [120] P. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *International Journal of Computer Vision*, vol. 50, no. 1, pp. 35–61, 2002. 139
- [121] P. J. Huber, *Robust Statistics*. John Wiley and Sons, 1981. 140

REFERENCES

- [122] E. Serradell, M. Özuysal, V. Lepetit, P. Fua, and F. Moreno-Noguer, “Combining geometric and appearance priors for robust homography estimation,” *European Conference on Computer Vision*, pp. 58–72, 2010. 140
- [123] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, “Bundle adjustment a modern synthesis,” in *Vision Algorithms: Theory and Practice, LNCS*. Springer Verlag, 2000, pp. 298–375. 147
- [124] M. Lourakis and A. Argyros, “Sba: A software package for generic sparse bundle adjustment,” *ACM Trans. Math. Softw.*, vol. 36, no. 1, pp. 1–30, 2009. 148
- [125] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, “BRIEF: Computing a Local Binary Descriptor Very Fast,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2012. 148