Exploiting Structural Regularities and Beyond: Vision-based Localization and Mapping in Man-Made Environments

Yi Zhou

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

January 2019

Except where otherwise indicated, this thesis is my own original work.

Yi Zhou
1 January 2019

To my beloved families.

Acknowledgments

To be honest, I feel incredible when I write down this page and prepare the documentation for the defence of a PhD degree.

I was supposed to be an artist if my vocal chords were not damaged during my voice changing period. The memory about my childhood before the age of thirteen is all about vocal training and a succession of shows and competitions. Therefore, I was a person of far more perceptual thinking than logistic thinking. When one door closes, another opens. I made two of my best friends (Han Gao and Yicheng Zhang) in college, who taught me basic skills of programming using C language. At the 3rd year of college, they brought me to visit the robotic lab (iTR Lab) in our university, where I was deeply impressed by an autonomous unmanned helicopter. Thanks to Beihang University for allowing students to choose their honour year project freely between departments, I joined the iTR lab in the final year and developed an algorithm that helped small unmanned helicopters to land on a target automatically using visual information. This was when and how I began to touch computer vision.

The incredible journey of my PhD began since the August of 2014. I would like to thank the Chinese Scholarship Council (CSC) for the financial support and Assoc/Prof. Hongdong Li, the chair of my supervisory panel for giving me the opportunity to study in one of the top computer vision and robotic groups in the world – Australian Centre of Robotic Vision (ACRV). The most impressive thing in the first year was the weekly meetings with Hongdong, Laurent and Yuchao. Their agile thinking and prudence made me realize how simple and naïve I was in this field. After one year's basic training, including literature review and attempts to several topics, I focused on the topic of visual odometry (VO) and simultaneous localization and mapping (SLAM). Though VO/SLAM is commonly believed to be a well studied topic from the perspective of theory, I still believe this topic is challenging for the following reasons. First, a modern SLAM system usually consists of many techniques in computer vision, from fundamental image processing, to geometry and to optimization methods, etc. Therefore, a qualified developer of SLAM systems needs to know well all the necessary techniques and have a good engineering ability to maintain such a big system. Second, creating things that work in practical cases is far more complicated than to prove it by simulation. To make systems robust and computationally efficient, researchers never stop exploring new features, constraints, and more efficient programming techniques on modern hardware. Walking on this tough road, I feel so lucky to have my primary supervisor, A/Prof. Laurent Kneip to be always with me. He is always patient and would like to share whatever he knows with me. I have learned so much from him, not only the way he thinks, but also the engineering skills that are keys to become a good researcher in robotic vision. The time we spent together to work towards deadlines will never be forgotten.

In the last year of my PhD, I was luckily invited by Prof. Davide Scaramuzza to visit the Robotic and Perception Group (RPG) — a joint lab affiliated with both the University of

Zurich (UZH) and ETH Zurich. There I saw how a group of smart people collaborated with each other and made things happen amazingly fast. My research at RPG is about developing a novel event-based mapping algorithm that exploits the full potential of event cameras. I would like to show my special appreciation to Prof.Davide Scaramuzza, Dr. Guillermo Gallego and Henri Rebecq, for their kind help in either research and life in Zurich. Besides, I feel deeply honoured to be awarded a fellowship grant by the Swiss National Centre of Competence in Research (NCCR) for my research on event-based reconstruction.

I also would like to show my appreciation to all my friends in the laboratory, for their sharing and giving. It is really my pleasure to meet you guys, Dr. Gao Zhu, Dr. Jiaolong Yang, Dr. Pan Ji, Dr. Juan David Adarve, Dr. Yifei Huang, Ilya Magafurov, YonHon Ng, Yiran Zhong, Liu Liu, Liyuan Pan, Jun Zhang, Jing Zhang, etc. The memory will stay in my mind forever.

Last but not the least, I am deeply grateful to my families for their love and great support. Special thanks go to my wife, my best friend and soul mate, Dr. Yi Yu, for her company and consideration all the time.

This research is supported by the Australian Centre for Robotic Vision. Yi Zhou acknowledges the financial support from the China Scholarship Council (CSC) for his PhD Scholarship No.201406020098.

Abstract

Image-based estimation of camera motion, known as visual odometry (VO), plays a very important role in many robotic applications such as control and navigation of unmanned mobile robots, especially when no external navigation reference signal is available. The core problem of VO is the estimation of the camera's ego-motion (*i.e.* tracking) either between successive frames, namely relative pose estimation, or with respect to a global map, namely absolute pose estimation. This thesis aims to develop efficient, accurate and robust VO solutions by taking advantage of structural regularities in man-made environments, such as piece-wise planar structures, Manhattan World and more generally, contours and edges. Furthermore, to handle challenging scenarios that are beyond the limits of classical sensor based VO solutions, we investigate a recently emerging sensor — the event camera and study on event-based mapping — one of the key problems in the event-based VO/SLAM. The main achievements are summarized as follows.

First, we revisit an old topic on relative pose estimation: accurately and robustly estimating the fundamental matrix given a collection of independently estimated homograhies. Three classical methods are reviewed and then we show a simple but nontrivial two-step normalization within the direct linear method that achieves similar performance to the less attractive and more computationally intensive hallucinated points based method.

Second, an efficient 3D rotation estimation algorithm for depth cameras in piece-wise planar environments is presented. It shows that by using surface normal vectors as an input, planar modes in the corresponding density distribution function can be discovered and continuously tracked using efficient non-parametric estimation techniques. The relative rotation can be estimated by registering entire bundles of planar modes by using robust L1-norm minimization.

Third, an efficient alternative to the iterative closest point algorithm for real-time tracking of modern depth cameras in Manhattan Worlds is developed. We exploit the common orthogonal structure of man-made environments in order to decouple the estimation of the rotation and the three degrees of freedom of the translation. The derived camera orientation is absolute and thus free of long-term drift, which in turn benefits the accuracy of the translation estimation as well.

Fourth, we look into a more general structural regularity — edges. A real-time VO system that uses Canny edges is proposed for RGB-D cameras. Two novel alternatives to classical distance transforms are developed with great properties that significantly improve the classical Euclidean distance field based methods in terms of efficiency, accuracy and robustness.

Finally, to deal with challenging scenarios that go beyond what standard RGB/RGB-D cameras can handle, we investigate the recently emerging event camera and focus on the problem of 3D reconstruction from data captured by a stereo event-camera rig moving in a static scene, such as in the context of stereo Simultaneous Localization and Mapping.

Key words: Visual Odometry (VO), Piece-wise Planar Environment, Manhattan World,

Distribution Alignment, RGB-D Camera, 3D-2D Registration, Distance Transform, Event Camera.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Contributions

Where am I? Who am I?
How did I come to be here?
What is this thing called the world ...

Søren Kierkegaard, Danish philosopher.

As one of the hypotheses that may explain the burst of apparently rapid evolution in the lower Cambrian, the "Light Switch" theory of Parker [2016] believes that the evolution of eyes started an arms race that accelerated evolution. The rationale behind this hypothesis is that vision adds context of a scene, which enables creatures to more easily recognize food, a potential mate, or a predator. This is definitely an evolutionary advantage. Now, sighted creatures cover millions of species, from a part of invertebrates, such as some insects, to almost all vertebrates, including humans. From some molecularly similar chemoreceptor cells to photoreceptor cells, the eye experiences a long history of evolution to become a dedicated organ (Nilsson [1996]). These special cells are very sensitive to light, more accurately photons. The signals of those photons are transmitted to the brain, where they are decoded as colors and shapes. Modern cameras work in the similar principle as our eyes do, whereas imaging chips (e.g. semiconductor charge-coupled devices (CCD) or active pixel sensors in complementary metal–oxide–semiconductor (CMOS)) play the role of photoreceptor cells. With these artificial eyes, researchers expect to endow robots the ability to perceive the world visually as we humans do. The missing part right here is the “algorithm” in robots’ brains to process the data.

When stepping into an unknown environment, like humans, the first priority of a robot is to be aware of where it is and what the surrounding environment is like. Vision based ego-motion estimation, coined as Visual Odometry (VO) by Nistér et al. [2004], has been an active field of research for more than three decades. It has wide application domains including augmented reality (AR) and autonomous driving, *etc.* Specifically, VO plays an essential role in the field of robotic control and navigation when no external reference signal is available. Examples are given by rovers operating on Mars (Moravec [1980]; Lacroix et al. [1999]), autonomous underwater vehicles (AUVs) carrying out exploration under the ocean (Corke et al. [2007]; da Costa Botelho et al. [2009]) and unmanned aerial vehicles (UAVs) patrolling in a GPS-denied environment, such as an indoor scene, a forest or an urban canyon (Courbon et al. [2009]; Tomic et al. [2012]; Forster et al. [2013]; Langelaan and Rock [2005]), *etc.* All of above systems need an alternative navigation modality which helps the robots to know their

own status of motion with respect to surrounding environments.

1.1 History of VO

The term VO was first used by Srinivasan et al. [1997] to define motion orientation of honey bees. This term has become popular in the field of computer vision and robotics since the paper *Visual Odometry* (Nistér et al. [2004]) was published. However, in fact the first known work implementing this idea is a stereo VO system by Moravec [1980] for NASA's Mars rover in 1980. The goal of the project is to give an alternative to wheel odometry which is always affected by slippage in uneven terrains. In the following two decades, the research on VO was led by NASA/JPL in preparation for the 2004 Mars mission (Matthies and Shafer [1987]; Matthies [1989]; Lacroix et al. [1999]; Olson et al. [2000]).

Stereo cameras are the most popular choice in the early solutions (Matthies and Shafer [1987]; Matthies [1989]; Lacroix et al. [1999]; Olson et al. [2000, 2003]; Cheng et al. [2005]; Milella and Siegwart [2006]; Howard [2008]). The reason is that stereo systems can recover 3D information without a prior of motion, which turns the estimation of the relative pose into a process of solving a straightforward 3D-to-3D point registration problem. Different motion estimation schemes are introduced by Nistér et al. [2004] and Comport et al. [2007]. Nistér et al. [2004] performed a 3D-to-2D point registration while Comport et al. [2007] relied on the quadrifocal tensor, which allowed motion estimation to be computed from 2D-to-2D image matches without having to triangulate 3D points.

An alternative to stereo based solutions is to use a single camera. The reason of interest in the monocular case is three-fold. First, stereo systems need to be calibrated intrinsically and extrinsically, which makes them more complicated compared to monocular systems. Second, multiple cameras will lead to additional energy consumption, which may not be available to some small mobile platforms. Last but not the least, a stereo system is known as to degenerate to the monocular case when the distance from the cameras to the scene becomes much bigger than the length of the baseline.

Some of existing works are worth special mentioning because they either created standards and inspired following works or still represent the state of the art.

1.1.1 Probabilistic Filter based Monocular SLAM

Mono-SLAM (Davison [2003]; Davison et al. [2007]) set the standard framework for traditional Bayesian filtering based visual SLAM framework. It uses image features to represent landmarks in the map and iteratively updates the probability density of features' depth by frame-to-frame matching and triangulation. A feature-based sparse map is created consequently and the full state vector including the robot's pose and 3-D locations of landmarks are updated within an EKF framework. Further pipelines worth mentioning, such as (Montemerlo et al. [2002]; Pupilli and Calway [2005, 2006]), utilized a particle filter instead of an EKF framework in camera/laser pose tracking. Compared to EKF based pipelines, these works scale better with the number of landmarks in the map.

1.1.2 Nister's VO

Nister *et al.* made several contributions to the field of VO. First, they presented an efficient five-point algorithm (Nistér [2004]) which would not degenerate in the case of coplanarity as the normalized eight-point algorithm (Hartley [1997]) does. Second, they provided a 3D-to-2D formulation of VO (Nistér et al. [2004]), which performs 3D reconstruction and camera pose estimation in an alternated fashion. More importantly, they concluded that the 3D-to-2D scheme is more accurate compared to either the 2D-to-2D or the 3D-to-3D scheme.

1.1.3 Parallel Tracking and Mapping (PTAM)

The standard of the front-end for modern SLAM systems is set by a work from the community of AR. Parallel tracking and mapping, also known as PTAM, was proposed by Klein and Murray [2007]. PTAM is a feature based method and follows the 3D-to-2D formulation described in (Nistér et al. [2004]). Its original design consists of decoupling the mapping from the tracking. This enables keyframe based bundle adjustment (BA) (Triggs et al. [1999]) that optimizes over many more points than the filter based method. Moreover, PTAM does not need to maintain an estimate of a dense covariance matrix as Mono-SLAM does, therefore much faster.

1.1.4 State of the Art

Based on the schemes of previous works, state-of-the-art systems keep making improvement from perspectives of both theory and system engineering. Based on the different information used for motion estimation, they could be clustered into four categories: feature based methods, direct methods, hybrid methods, and methods based on 3D point set registration.

- **Feature based methods:**

  The front-end of ORB-SLAM (Mur-Artal et al. [2015] is a feature-based method and is quite similar to PTAM. It achieves better performance compared to PTAM by 1) extending an additional thread for loop closing which guarantees globally consistent localization and mapping; 2) automatically initializing the map via selecting a model between the Homography and the Fundamental matrix, while PTAM requires manual operation to finish the initialization; 3) utilizing ORB features (Rublee et al. [2011]) instead of image patches used in PTAM which improves matching accuracy under scale and orientation changes; 4) multi-scale mapping which consists of a local graph for pose refinement, a co-visibility graph for local bundle adjustment, and an essential graph for global bundle adjustment after a loop closing is detected and verified.

- **Direct methods:**

  Different from feature based methods, direct methods utilize the intensity information of the whole image for motion estimation, which have been proven to be more effective in textureless environments. Among direct methods, DTAM (Newcombe et al. [2011b]) is the first real-time system which is able to estimate a dense depth map at each keyframe.

However, processing every pixel over all image sequences is very computationally expensive which leads to dependency on GPU hardwares. The appearance of low cost RGB-D cameras eliminates the estimation of depth maps, which makes it applicable for running fully dense methods on CPU in real time. Kerl *et al.* proposed a dense visual odometry (DVO) (Kerl et al. [2013a]). The relative pose between two frames is estimated by solving a 3D-2D registration problem. A probabilistic formulation is given for improving the robustness in the presence of outliers and sensor noises. More recently, Engel *et al.* proposed a semi-dense visual odometry (SDVO) using a monocular camera (Engel et al. [2013]). SDVO only uses pixels that are along the boundary of structures. Therefore, it is able to run in real-time on a CPU. SDVO utilizes the same tracking method as DVO does. The contribution of SDVO is the probabilistic fusion based mapping, in which measurement uncertainties originating from both geometric and photometric cues are effectively modelled. More recently, Engel et al. [2017] present the direct sparse odometry (DSO), which applies direct method on a number of sparse points. These sparse points are sampled across image areas with sufficient intensity gradient. To deal with imperfect brightness constancy, Engel et al. [2017] propose a photometric calibration pipeline, which recovers the irradiance images and therefore increases the tracking accuracy.

- **Hybrid methods:**

  Direct methods are known to be sensitive to inconsistent illumination, while feature-based approaches fail when not enough textures appear. Hybrid methods have claimed a better performance in these challenging cases. For example, Scaramuzza and Siegwart [2008] used the image appearance to estimate the rotation of the car and features from the ground plane to estimate the translation and the absolute scale. Forster et al. [2014] used the photometric information around sparse features, which leads to precise, robust and amazingly fast performance (400 fps on a laptop).

- **3D point registration based methods:**

  3D point set registration is a traditional problem that has been investigated extensively in the computer vision community. We are limiting the discussion to methods that process mainly rigid, geometric information. The most commonly used method is the ICP algorithm proposed by Besl and McKay [1992], which performs registration through iterative minimization of the SSD distances between spatial neighbours in two point sets. Classical ICP methods are prone to local minima once the displacement is too large. In order to tackle situations of large view-point changes, Yang et al. [2013] investigated globally optimal solutions to the point set registration problem. This method is however inefficient and thus not suited for real-time applications, where the frame-to-frame displacement remains small enough for a successful application of local methods. From a more modern perspective, the ICP algorithm and its close derivatives (Pomerleau et al. [2011]; Newcombe et al. [2011a]; Whelan et al. [2012a]; Pomerleau et al. [2013]) still represent the algorithm of choice for real-time tracking of depth sensor.

1.2 Motivation and Objectives — Exploiting Structural Regularities and Beyond

By looking at the history of VO, we see that a lot of efforts have been made in order to achieve efficient pose estimation while keeping the global drift as small as possible. Among the massive body of existing work, few of them focus on improving the tracking performance via taking advantage of the structural prior hidden in structural regularities of man-made environments.

As a frequently occuring geometric regularity in man-made environments, planar structures provide a strong geometric constraint on ego-motion estimation. In the 2D relative pose problem, the co-planarity of points leads to a compact expression of motion and structure — the homography. It is believed that a prior on the structure would benefit the motion estimation and vice versa (Szeliski and Torr [1998]), thus, the estimation of the fundamental matrix would ideally benefit from using homographies as input. When 3D information is available, the piece-wise planar environment enables us to create alternative solutions to ICP, which is computationally expensive and suffers from local minimums. We show that the surface normal vectors of those planes can be used to efficiently estimate the relative rotation between different perspectives. Moreover, if the environment consists of three dominant planes that are orthogonal to each other, namely a Manhattan World (MW), the rotation estimation is globally drift-free while each degree of translational freedom could be further solved in parallel. A more general structure regularity is the contours/edges. They typically correspond to pixels with strong intensity gradient in images. Pose estimation based on these pixels are less affected when the photometric consistency assumption does not strictly hold. We can see that both the efficiency and the accuracy benefits from exploiting structural regularities in this research. However, when application scenarios bring in challenging conditions such as high-speed motion, high dynamic range (HDR) that are beyond what normal sensors (standard RGB/RGB-D cameras) can handle, existing solutions are no longer applicable. Accordingly, it is imperative to investigate recently emerging sensors and to develop novel algorithms that fit their distinct characteristics.

1.2.1 2D Geometrically Constrained Relative Pose Estimation: Points on Planes

The epipolar geometry of two perspective images, illustrated by Fig. 1.1, demonstrates that a 3D point $\mathbf{X}$ observed in one image must lie (when no occlusion exists) on the epipolar line in the other image. This constraint can be described by a singular 3×3 matrix. When the camera is calibrated, the matrix is known as the essential matrix $\mathbf{E}$. For uncalibrated systems, it is known as the fundamental matrix $\mathbf{F}$. The estimation of the fundamental matrix is a classical and thoroughly studied topic which plays an essential role in many applications involving multiple-view geometry.

The most popular method for estimating the fundamental matrix is to solve Eg. 1.1 given sparse correspondences between local invariant keypoints, for instance SIFT features (Lowe [2004]).

$$\mathbf{x}'\mathbf{F}\mathbf{x} = 0 \tag{1.1}$$

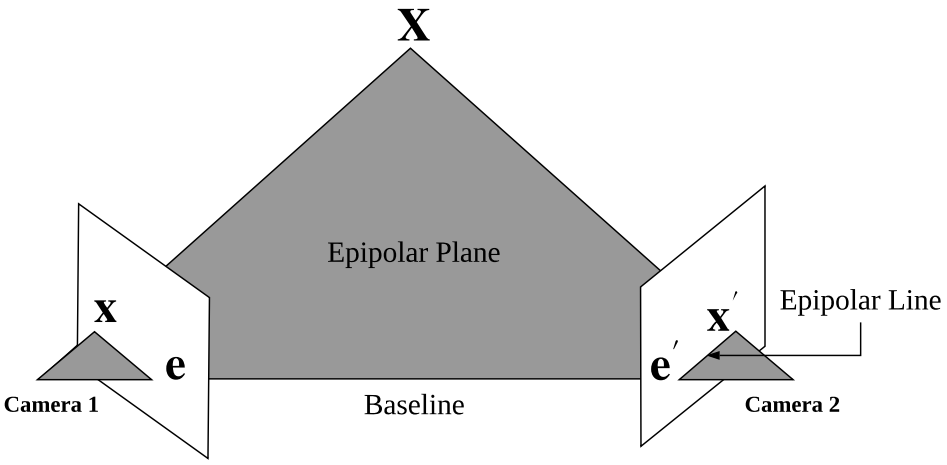


Figure 1.1: Illustration of the epipolar geometry. The optical centres of the two cameras and the 3D point **X** determines the epipolar plane. Epipoles **e**, **e**′ are defined as the intersection points of the baseline with each image plane. The epipolar plane and an image plane intersects at an epipolar line.

Using the direct linear transform (DLT), Eq 1.1 is transformed into a linear equation system

$$\mathbf{Af} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}, \tag{1.2}$$

where $(x_i, y_i) \leftrightarrow (x'_i, y'_i), i = 1 \cdots n$, denotes the feature correspondences, and **f** consists of all entries of the fundamental matrix **F**. Seven points constitute the minimal configuration because the fundamental matrix has 7 degrees of freedom (DoF). Compared to the eight-point algorithm, the seven point algorithm needs an additional step to calculate the linear combination factor of the obtained two-dimensional null-space. While seven point correspondences represent the minimum for estimating the fundamental matrix (Stewart [1999]), the eight-point algorithm (Longuet-Higgins [1987]) is the most widely used method because of its linear nature and thus simplicity to implement. However, it was only after Hartley published his seminal work (Hartley [1997]) on using data normalization that the eight-point algorithm became truly useful in practice.

When a point is on a plane, its projections on two images can be associated by not only a fundamental matrix, but also a homography. The association is denoted by Eq. 1.3 and the geometry is illustrated in Fig. 1.2.

$$\mathbf{x}' = \mathbf{Hx} \tag{1.3}$$

It is believed that the estimation of both structure and motion can be improved by incorporating additional geometric constraints like coplanarity of certain points. Luong and Faugeras [1993, 1996] are the first who proposed to estimate the fundamental matrix with multiple homographies in a linear way. They compared their linear solutions with other non-linear ones concluding that none of the developed methods is stable under noise. In other words, though the direct linear method is quite simple and straightforward, it has limited practical usefulness.

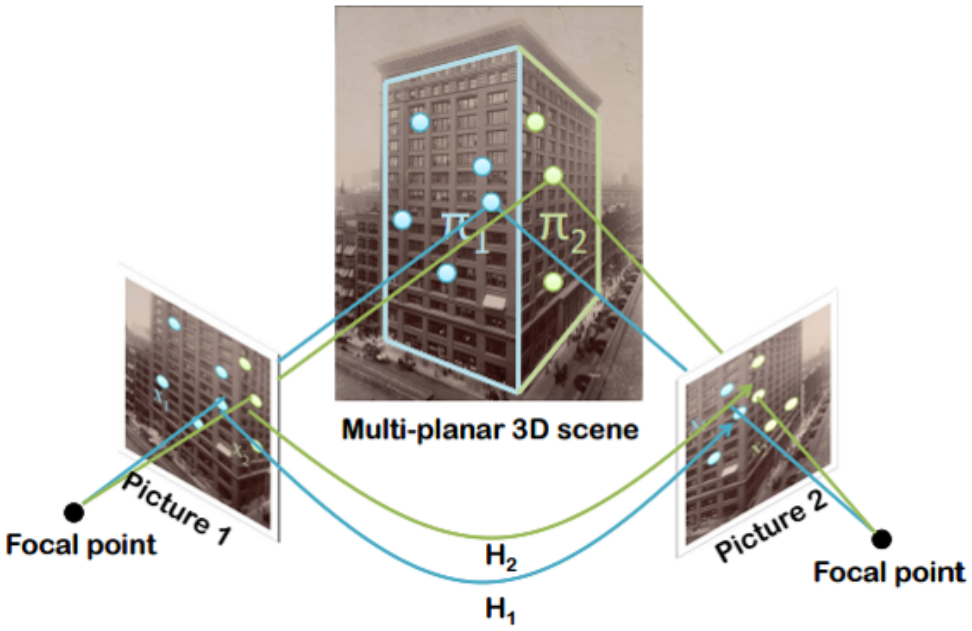


Figure 1.2: Illustration of a hompgrahy that associates the projections of a point on a planar structure. The raw image is from (Szpak et al. [2014]).

Zhang [1998] gave a thorough review on the techniques of fundamental matrix estimation and its uncertainty. The bad performance of the Direct Linear Transformation (DLT) applied to the compatibility relation between the homography $\mathbf{H}$ and the fundamental matrix $\mathbf{F}$ was however not discussed in much detail. Szeliski and Torr [1998] thoroughly discussed three methods used for solving structure from motion (SfM) with planes. They presented an analysis of the robustness of each method and then suggested to estimate the fundamental matrix with hallucinated points (HP) that lie on planes instead of using the compatibility equation (and thus the homograhies directly). Agarwal et al. [2005] demonstrated that the compatibility constraint is an implicit equation in $\mathbf{H}$ and $\mathbf{F}$. They also concluded that an explicit expression like $\mathbf{F} = [\mathbf{e}']_\times \mathbf{H}$ is more suitable for a computational algorithm. Vincent and Laganiére [2001] proposed a detection algorithm for planar homographies working on a pair of uncalibrated images. They claimed that the estimation of the fundamental matrix from point correspondences derived from homographies allows to use data normalization techniques, and thus performs much better than using the homographies directly. A method was introduced to estimate the fundamental matrix with a homology by Sinclair et al. [1995]; Pritchett and Zisserman [1998]; Hartley and Zisserman [2003]. Theoretically, a homology has two identical eigenvalues and another unique one which is corresponding to the epipole $\mathbf{e}'$. However, in practical cases, the imperfect homographies may lead to complex eigenvalues. It is hard to choose which eigenvalue corresponds to the unique one; the real parts of the eigenvalues are often equally spread and/or very close to each other.

1.2.2 Tracking a 3D Sensor in Piece-wise Planar Environments and Manhattan Worlds

3D sensors typically refer to devices that are able to obtain the 3D information of a scene, especially those that provides dense 3D measurements, such as Velodyne LiDARs, Microsoft Kinects and so on. The problem of tracking a 3D sensor is typically formulated as a point cloud registration problem, shown as Fig. 1.3. A 3D point cloud can be denoted as $\mathcal{P} = \mathbf{p}_i, i = 1, ..., M$, where $\mathbf{p}_i$ represents the coordinate of a 3D point, and M is the number of 3D points in the point cloud. Assume a 3D camera captures a scene at two different perspectives and generates two point clouds $\mathcal{P}, \mathcal{Q}$. Since the coordinate of the point cloud is described in the camera coordinate system, to estimate the relative pose between two different perspectives is equivalent to rigidly registering the two point clouds. The point cloud registration problem is generally written as

$$\mathbf{R}, \mathbf{t} = \arg\min_{\mathbf{R},\mathbf{t}} \Phi(G(\mathcal{P}, \mathbf{R}, \mathbf{t}), \mathcal{Q}), \tag{1.4}$$

where $(\mathbf{R}, \mathbf{t}) \in SE(3)$, function G applies rigid transformation $(\mathbf{R}, \mathbf{t})$ on $\mathcal{P}$, and $\Phi(\cdot, \cdot)$ measures the registration error in a certain metric. The most commonly used way to solve problem 1.4 is the ICP algorithm (Besl and McKay [1992]), in which the registration error in 1.4 is defined as the sum of the squared closest-point distances,

$$\mathbf{R}, \mathbf{t} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{M} \min_{j=1,\cdots,N} \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_j\|^2, \tag{1.5}$$

where $\mathbf{p}_i, i = 1, \cdots, M$ and $\mathbf{q}_j, j = 1, \cdots, N$ denote the 3D points in $\mathcal{P}$ and $\mathcal{Q}$, respectively. The ICP algorithm performs registration through iterative minimization of the sum of the squared closest-point distances between spatial neighbours in two point sets.

In order to avoid the costly repetitive derivation of point-to-point correspondences, the community has also investigated the representation and alignment of point clouds using density distribution functions. The idea was proposed by Chui and Rangarajan [2000a] and Tsin and Kanade [2004], who represented point clouds as explicit Gaussian Mixture Models (GMM) or implicit Kernel Density Estimates (KDE), and then found the relative transformation (not necessarily Euclidean) by aligning those density distributions. Jian and Vemuri [2011] summarized the idea of using GMMs for finding the aligning transformation, and notably derived a closed-form expression for computing the L2 distance between two GMMs. Yet another alternative which avoids the establishment of point-to-point correspondences was given by Fitzgibbon [2003], who utilizes a distance transformation in order to efficiently and robustly compute the cost of an aligning transformation. The distance transformation itself, however, is again computationally intensive.

Classical ICP or even density alignment based methods are prone to local minima once the displacement becomes too large and thus also the point cloud structure is subjected to intensive changes. In order to tackle situations of large view-point changes, the community has therefore investigated globally optimal solutions to the point set registration problem, such as Yang et al. [2013]. These methods are however inefficient and thus not suited for real-time applications, where the frame-to-frame displacement anyway remains small enough for a

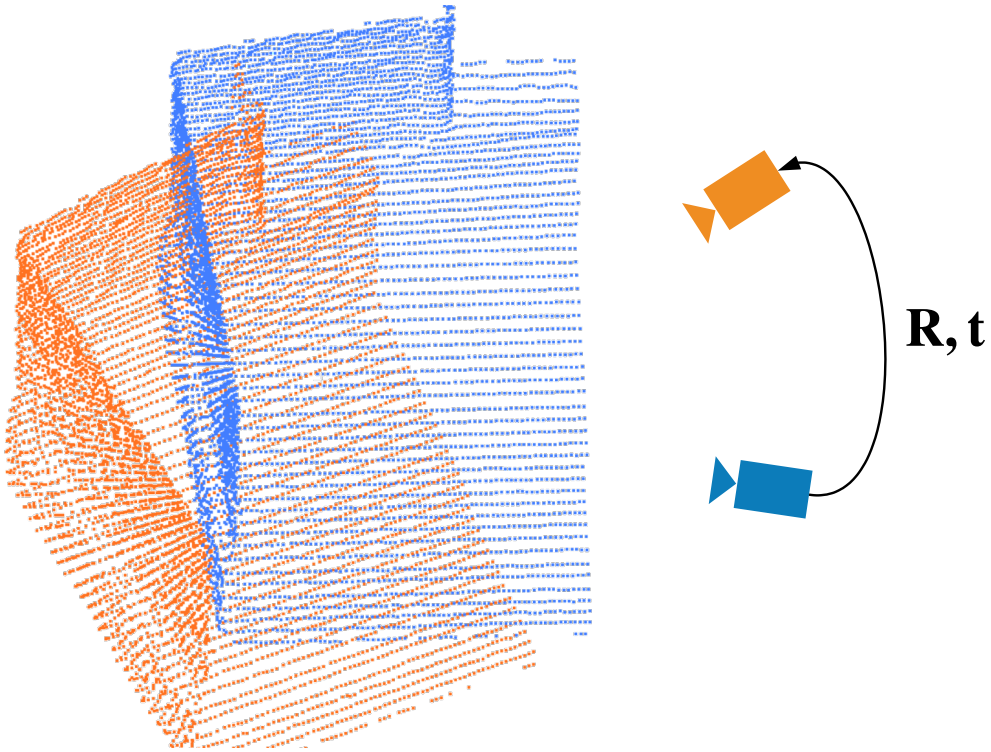
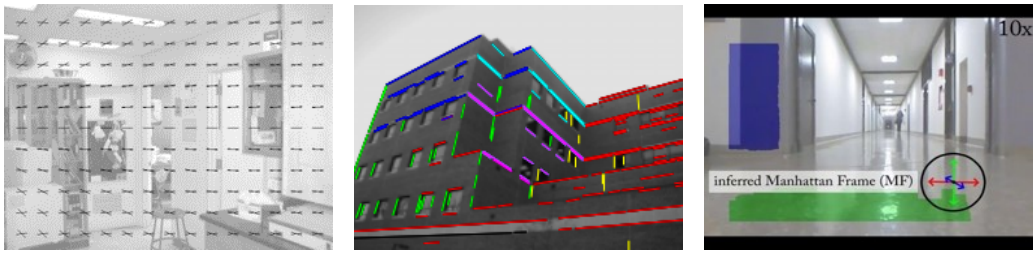


Figure 1.3: Illustration of point cloud registration problem. The point cloud is from (Sanchez et al. [2017]). The 3D camera and its observation are associated by using the same color.

successful application of local methods.

Planar structures in man-made environments can benefit both tracking and mapping performance when using exteroceptive sensors. Weingarten and Siegwart [2006] and Trevor et al. [2012] used planar-segment features extracted from 3D sensors. Both of them claimed that the planar segments can be used to improve the data association, which in turns benefits the whole localization and mapping process. Taguchi et al. [2013] combined point and plane features towards fast and accurate 3D registration. From exterior side to interior side, modern buildings frequently contain orthogonal structures in the surface arrangement. The property was coined as *Manhattan World* (MW) in (Coughlan and Yuille [1999]), where they formulated vanishing point estimation from a single RGB image as a Bayesian inference problem. Košecká and Zhang [2002] presented a video compass using a similar idea. Tracking the *Manhattan Frame* can be regarded as absolute orientation estimation, and thus leads to a significant reduction or even complete elimination of the rotational drift. Silberman et al. [2012] improved MW orientation estimation by introducing depth and surface normal information obtained from 3D sensors. More recently, Straub et al. [2014] proposes the inference of an explicit probabilistic model to describe the world as a mixture of Manhattan frames. They employ an adaptive Markov-Chain Monte-Carlo sampling algorithm with Metropolis-Hasting split/merge moves to identify von-Mises-Fisher distributions of the surface normal vectors. In (Straub et al. [2015a]), they adapted the idea to a more computationally friendly approach for real-time tracking of a single, dominant MF. As illustrated in Fig. 1.4, most of the existing works are limited to the

estimation of the rotation of a camera through tracking the *Manhattan Frame*. Few have shown how to perform a full 6-Dof motion estimation based on the Manhattan assumption.



(a) Coughlan and Yuille [1999]. (b) Košecká and Zhang [2002]. (c) Straub et al. [2015a].

Figure 1.4: Existing works that estimate the motion (rotation) of a camera by taking advantage of the Manhattan World assumption.

1.2.3 One Step Beyond: A More General Regularity — Edges

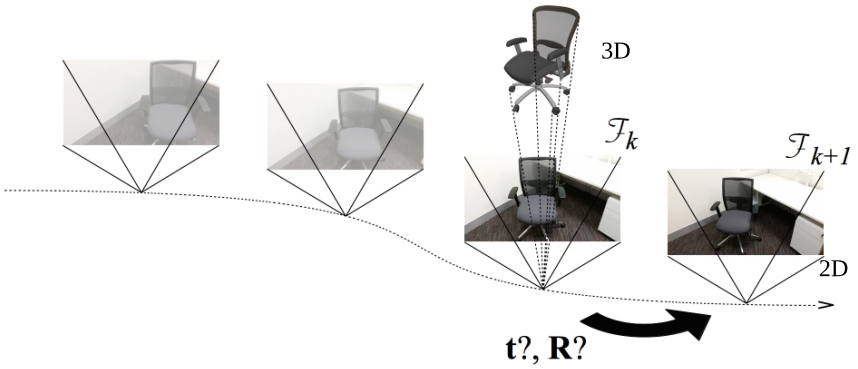
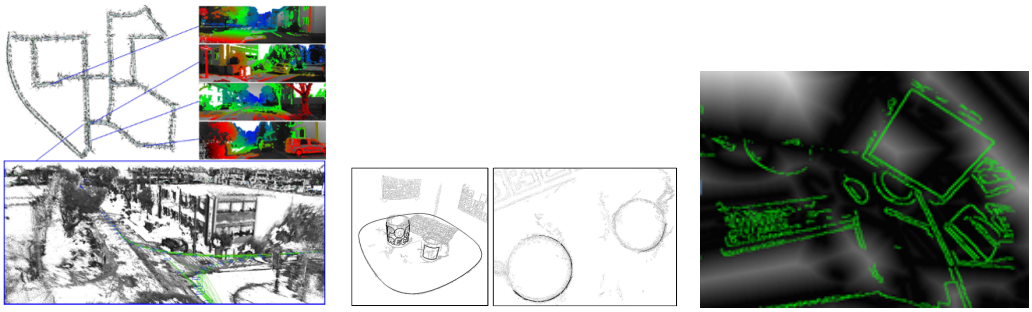


Figure 1.5: Illustration of the 3D-2D registration problem. The goal is to estimate the relative pose of frame $\mathcal{F}_{k+1}$ (with 2D information) w.r.t frame $\mathcal{F}_k$ (with 3D information).

From a more general perspective, either the piece-wise planar environment or the Manhattan World assumption is so strong that the derived VO systems cannot be applied to environments where those assumptions do not sufficiently hold. To move one step beyond, a more general structural regularity is required. Edges are abundant in man-made environments. They can be boundaries of either structures or textures. Any approach that replies on edges is usually reported to be outperforming sparse feature based methods in textureless environments and to be less affected by inconsistent illumination than direct methods. Several typical pipelines are shown in Fig. 1.6 As a special case of edges, lines have been used as alternative features to points and widely employed in many VO and SLAM frameworks such as (Eade and Drummond [2009]; Lu and Song [2015]). One reason is that line features are easily parametrized and included into a bundle adjustment (BA) framework for the purpose of global optimization (Eade and Drummond [2009]; Klein and Murray [2008]). However, straight lines are not general features because contours of objects can be arbitrary curves in 3D space. Therefore,



(a) Engel et al. [2013, 2014]. (b) Nurutdinova and Fitzgibbon [2015]. (c) Kuse and Shen [2016].

Figure 1.6: Examples that use boundaries, edges and semi-dense regions around them.

Nurutdinova *et al.* presented a method which uses parametric curves as landmarks for motion estimation and BA (Nurutdinova and Fitzgibbon [2015]). In contrast, non-parametric methods are more popular. Engel *et al.* applied direct photometric registration to semi-dense regions defined as all neighbouring pixels of edges (Engel et al. [2013, 2014]). A more relevant work to this thesis is (Kuse and Shen [2016]), which presents a direct edge alignment approach for 6-DOF tracking. Non-parametric methods are always formulated as a 3D-2D registration problem, as illustrated by Fig. 1.5. When using photometric measurements (Newcombe et al. [2011b]; Engel et al. [2013, 2014]; Schoeps et al. [2014]), the objective function is written as,

$$\mathbf{R},\mathbf{t} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_i (I_{k+1}(W(\mathbf{x}_i, z_i; \mathbf{R},\mathbf{t})) - I_k(\mathbf{x}_i))^2 \tag{1.6}$$

where $I(\cdot)$ returns the intensity of a given pixel coordinate, $W(\cdot)$ warps a pixel under its depth z_i and the optimized motion parameters $\mathbf{R},\mathbf{t}$. The optimal motion leads to the global minimum of the objective function. When the residuals are measured in geometric distance (Kneip et al. [2015]; Kuse and Shen [2016]; Zhou et al. [2017]), the objective function is denoted as,

$$\mathbf{R},\mathbf{t} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_i \mathcal{D}(W(\mathbf{x}_i, z_i; \mathbf{R},\mathbf{t})) \tag{1.7}$$

where $\mathcal{D}(\cdot)$ returns the distance to the closest point. The objective function is typically solved as an 3D-2D ICP problem, which needs to repeatedly search for the closest point for each warping pixel in each iteration. To accelerate the closest point searching, *distance transform* (DT) is introduced by Felzenszwalb and Huttenlocher [2004]. A look-up table under a certain distance metric is created to avoid repeated searching.

Geometric methods have a larger convergence basin than photometric methods, thus perform better when registering two frames under big transformation. As reported by Kuse and Shen [2016], classical 3D-2D registration pipelines that are solved using the Gauss-Newton method have no guarantee for convergence. This attributes to the fact that the objective function in Eq. 1.7 is not smooth. Kuse and Shen [2016] solved this problem by using the sub-gradient method. The nature of ICP determines that the geometric 3D-2D registration pipelines are prone to local minimum, especially in the case of big motion or partial occlusion. To deal with

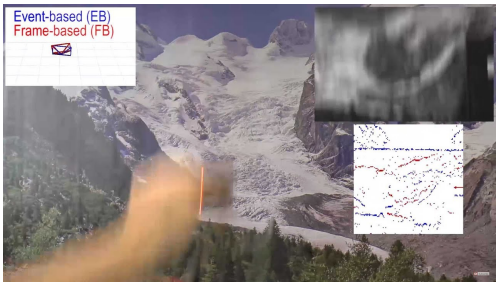
big motion, a coarse-to-fine strategy is typically used (Kuse and Shen [2016]). Nurutdinova and Fitzgibbon [2015] overcame the partial occlusion problem by using parametric curves while increasing the dimension of unknowns. Furthermore, robust tracking requires to handle outliers and noises effectively. Most of the exiting pipelines just empirically choose robust weight function rather than looking into the real probabilistic characteristics of the sensor.

1.2.4 Beyond the Limits: Event-based VO

When applying VO/SLAM techniques in practical applications, more complicated and challenging scenarios than what we see in the laboratory could emerge. An example is illustrated in Fig 1.7, in which a small UAV is performing a rescue task in a post-disaster environment. Fast operations are very much expected to save more lives. Therefore, image blur induced by high-speed motions must be handled. Besides, the drone may fly from indoor to outdoor, which can lead to severe illumination changes. In extreme cases, over/under-exposure can make that system temporarily blind. Thus, an ideal sensor needs to be capable of dealing with high dynamic range (HDR) scenarios. Each of these challenging factors can easily fail existing VO/SLAM solutions developed for traditional visual sensors. Accordingly, it is imperative to investigate recently emerging sensors and to develop novel algorithms that fit their distinct characteristics.

(a) An outdoor case.

(b) An indoor case.



(c) High-speed motion.

(d) High dynamic range.

Figure 1.7: Illustration of challenging scenarios for classical vision based navigation. Images are from http://rpg.ifi.uzh.ch/gallery.html.

Event cameras, such as the Dynamic Vision Sensor (DVS) (Lichtsteiner et al. [2008]), are novel devices that output pixel-wise intensity changes (called "events") asynchronously, at the time they occur. As opposed to standard cameras, they do not acquire an entire image frame at the same time, nor do they operate at a fixed frame rate. An illustration of the event camera's working mechanism is given in Fig. 1.8. The output of an event camera is a 3D spatial-temporal signal, which is typically denoted as a tuple,

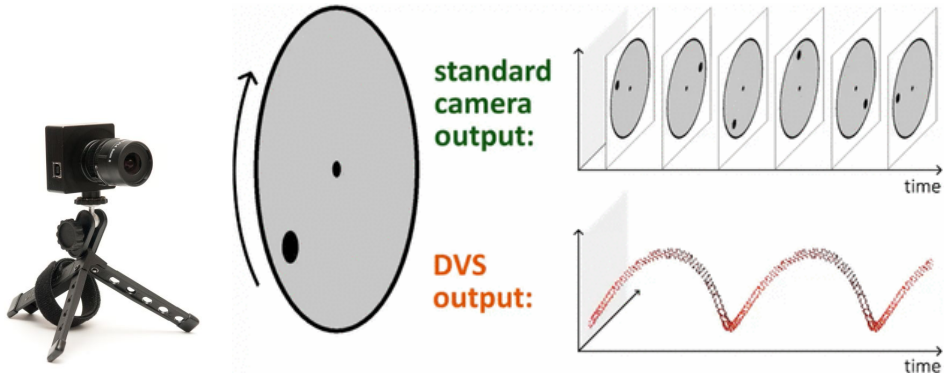
$$\mathbf{e} = \{u, v, t, p\}, \tag{1.8}$$

where u, v denotes the event coordinate on the sensor plane, t gives the timestamp when the event occurs and p tells the sign of the intensity change. This asynchronous and differential principle of operation reduces power and bandwidth requirements drastically. Endowed with microsecond temporal resolution, event cameras are able to capture high-speed motions, which would typically cause severe motion blur with standard cameras. In addition, event cameras have a very high Dynamic Range (HDR) (*e.g.* 140 dB compared to 60 dB of most standard cameras), which allows them to be used under a broad range of illumination. Hence, event cameras open the door to tackle challenging scenarios that are inaccessible to standard cameras, such as high-speed and/or HDR tracking (Mueggler et al. [2014]; Lagorce et al. [2015]; Zhu et al. [2017]; Gallego et al. [2017]), control (Conradt et al. [2009]; Delbruck and Lang [2013]) and Simultaneous Localization and Mapping (SLAM) (Kim et al. [2016]; Rebecq et al. [2017c]; Rosinol Vidal et al. [2018]).

The main challenge in visual processing with event cameras is to devise specialized algorithms that can exploit the temporally asynchronous and spatially sparse nature of the image data produced by DVS cameras, hence unlocking their full potential, whereas existing computer vision algorithms designed for conventional cameras do not directly apply in general. Some preliminary works on DVS addressed this issue by combining event cameras with other sensors, such as standard cameras (Censi and Scaramuzza [2014]; Kueng et al. [2016]) or depth sensors (Censi and Scaramuzza [2014]; Weikersdorfer et al. [2014]), in order to simplify the task at hand. Although this approach obtained certain success, the true potential of an event camera has not been fully exploited since parts of such combined systems are limited by the lower dynamic range devices.

1.3 Thesis Outline and Contributions

In Chapter. 2, we look into the 2D geometrically constrained relative pose estimation in piecewise planar environments. More specifically, we focus on answering a classical geometry question – how to determine the fundamental matrix from a collection of inter-frame homographies. The compatibility relationship between the fundamental matrix and any of the ideally consistent homographies can be used to compute the fundamental matrix. Using the direct linear transformation (DLT), the compatibility equation can be translated into a least squares problem and can be easily solved via SVD decomposition. However, this solution is extremely susceptible to imaging noise, hence rarely used. Inspired by the normalized eight-point algorithm, we show that a relatively simple but non-trivial two-step normalization of the input homographies achieves the desired effect, and the results are at last comparable to the less at-



(a) Event camera. (b) Working mechanism.

Figure 1.8: Illustration of an event camera and its working mechanism. Unlike standard RGB cameras that capture the scene at a fixed frame rate, the event camera only reports "events" — intensity changes. The images are from (Rebecq et al. [2017b]).

tractive hallucinated points based method. The algorithm is theoretically justified and verified by experiments on both synthetic and real data.

In Chapter. 3, an efficient 3D rotation estimation algorithm for depth cameras in piece-wise planar environments is presented. It shows that by using surface normal vectors as an input, planar modes in the corresponding density distribution function can be discovered and continuously tracked using efficient non-parametric estimation techniques. The relative rotation from the reference view to the current view can be estimated by registering entire bundles of planar modes. Robustness of the bundle registration process is achieved by performing a general ℓ_1-norm regression instead of simply solving a least-squares problem. Piece-wise drift-free performance is achieved as long as no bundle updates happen.

In Chapter. 4, we reply on the results of Chapter. 3 and make the further assumption about the environment that three mutually orthogonal planes exit. A highly efficient motion estimation framework is presented for 3D sensors such as the Microsoft Kinect v.2, based on alignment of density distribution functions. Absolute rotation is estimated by exploiting the properties of Manhattan Worlds, thus resulting in a manifold-constrained multi-mode tracking scheme. The individual translational degrees of freedom is efficiently estimated through 1D kernel density estimates. A real-time implementation is given, which is able to process dense depth images with VGA resolution at more than 50Hz on a CPU.

In Chapter. 5, we investigate a more general structural regularity — edges (arbitrary 3D curves) of structures and present a robust VO algorithm for RGB-D cameras. The method tracks the camera's 6 DoF motion with a 3D-2D geometric curve registration approach. Instead of using the classical Euclidean distance field, two novel alternatives are presented. The resulting method does not depend on bilinear interpolation, and enables adaptive sampling, parallel computation, and is capable of eliminating the registration bias. To improve robustness against noise and outliers, the ICP-based pipeline is formulated as a maximum a posteriori problem, which is subsequently transformed into a weighted least squares problem and solved with IRLS. We study the statistical properties of the sensor model, which leads to the optimal

choice from various robust M-Estimators. The proposed method outperforms state-of-the-art edge-alignment based method in terms of accuracy, efficiency and robustness.

In Chapter 6, to deal with challenging scenarios that are beyond what standard RGB/RGB-D cameras can handle, we investigate a recently emerging sensor — the event camera — and focus on the problem of 3D reconstruction from data captured by a stereo event-camera rig moving in a static scene, such as in the context of stereo Simultaneous Localization and Mapping. The proposed method consists of the optimization of an energy function designed to exploit small-baseline spatio-temporal consistency of events triggered across both stereo image planes. To improve the density of the reconstruction and to reduce the uncertainty of the estimation, a probabilistic depth-fusion strategy is also developed. The resulting method has no special requirements on either the motion of the stereo event-camera rig or on prior knowledge about the scene. Experiments demonstrate the proposed method can deal with both texture-rich scenes as well as sparse scenes, outperforming state-of-the-art stereo methods based on event data image representations.

1.3.1 Publication

The thesis is mainly based on the following publications during my PhD:

- Zhou et al. [2015] Y. Zhou, L. Kneip and H. Li, "**A Revisit of Methods for Determining the Fundamental Matrix with Planes**," 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Adelaide, SA, 2015, pp. 1-7.

- Kneip et al. [2015] L. Kneip, Y. Zhou and H. Li. "**SDICP: Semi-Dense Tracking based on Iterative Closest Points**". In Proceedings of the British Machine Vision Conference (BMVC), pages 100.1-100.12. BMVA Press, September 2015.

- Zhou et al. [2016a] Y. Zhou, L. Kneip and H. Li, "**Real-time rotation estimation for dense depth sensors in piece-wise planar environments**," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, 2016, pp. 2271-2278.

- Zhou et al. [2016b] Y. Zhou, L. Kneip, C. Rodriguez, and H. Li. "**Divide and conquer: Efficient density-based tracking of 3D sensors in Manhattan worlds**." In Asian Conference on Computer Vision (ACCV), pp. 3-19. Springer, Cham, 2016.

- Zhou et al. [2017] Y. Zhou, L. Kneip and H. Li, "**Semi-dense visual odometry for RGB-D cameras using approximate nearest neighbour fields**," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 6261-6268.

- Zhou et al. [2018b] Y. Zhou, H. Li, L. Kneip, "**Canny-VO: Visual Odometry with RGB-D Cameras based on Geometric 3D-2D Edge Alignment**," accepted by IEEE T-RO (published as Early Access by far).

- Zhou et al. [2018a] Y.Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, D. Scaramuzza, "**Semi-Dense 3D Reconstruction with a Stereo Event Camera**," European Conference on Computer Vision. Vol. 2. Springer, Cham, 2018.

Chapter 2

2D Geometrically Constrained Relative Pose Estimation: Points on Planes

In this chapter, we look into the problem of relative pose estimation in piece-wise planar environments. More specifically, we focus on answering a classical visual geometry question – how to determine the fundamental matrix from a collection of inter-frame homographies (more than two). The compatibility relationship between the fundamental matrix and any of the ideally consistent homographies can be used to compute the fundamental matrix. Using the direct linear transformation (DLT), the compatibility equation can be translated into a least squares problem and easily solved via SVD decomposition. However, this solution is extremely susceptible to noise and motion inconsistencies, hence rarely used. Inspired by the normalized eight-point algorithm, we show that a relatively simple but non-trivial two-step normalization for the input homographies achieves the desired effect, and the results are at least comparable to the less attractive hallucinated points method. The algorithm is theoretically justified and verified by experiments on both synthetic and real data.

2.1 Related Work — Three Classical Methods

Szeliski and Torr discussed three methods that can be used for the estimation of the fundamental matrix given several ($\geqslant 2$) homographies in (Szeliski and Torr [1998]), which are reviewed in the following.

- **Hallucinating additional correspondences:**

  Hallucinated points refer to augmented sample points on planes. Theses points are also called virtual control points. Hallucinated correspondences are generated by first creating several virtual 2D points $\mathbf{x}$ on image one which are assumed to be the projection of virtual points on the plane. Their corresponding points $\mathbf{x}'$ are then found by applying the corresponding homography to points $\mathbf{x}$. Then the fundamental matrix $\mathbf{F}$ is computed by applying normalized 8-point algorithm on the obtained hallucinated correspondences.

- **Direct linear method:**

The implicit compatibility relationship between inter-frame homographies and the fundamental matrix can be directly used for computing the fundamental matrix. The compatibility equation $\mathbf{F}^T\mathbf{H} + \mathbf{H}^T\mathbf{F} = \mathbf{0}$ gives six constraints (Luong and Faugeras [1993]) (for which only 5 are linearly independent). Therefore, at least 2 homographies are needed for computing the fundamental matrix. The question can be translated to a least squares problem by DLT and can be easily solved by SVD decomposition. However, this straightforward method is unstable for inaccurate homographies, sometimes leading to completely meaningless results. The reason given by Szeliski and Torr is that using the compatibility equation directly corresponds to sampling homographies at locations where their predictive power is very weak. The samples are far from having the normal distribution required for total least squares to work reasonably well.

- **Plane plus parallax:**

  Plane plus parallax techniques are always used to recover the depth (projective or Euclidean) of the scene. To compute the fundamental matrix, one of the homographies is chosen and used to warp all the feature points to the current frame. The epipole $\mathbf{e}'$ is computed by minimizing the sum of the weighted distance between the epipole and lines passing through corresponding points $\mathbf{x}_i$ and $\mathbf{x}'_i$. Then the fundamental matrix $\mathbf{F}$ can be computed by $\mathbf{F} = [\mathbf{e}']_\times \mathbf{H}$. This method cannot work well when points are evenly distributed over several planes. The computation is also more complicated and expensive compared to the former two methods.

2.2 A Robust Two-Step Linear Solution

The compatibility equation $\mathbf{F}^T\mathbf{H} + \mathbf{H}^T\mathbf{F} = \mathbf{0}$ gives only 6 linear equations (Luong and Faugeras [1993]). In fact, as shown later, only 5 of them are independent. Therefore, at least 2 homographies are needed to compute the fundamental matrix. Applying the DLT transformation to the compatibility equation leads to the least squares problem,

$$\mathbf{A}\mathbf{f} = \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_n \end{pmatrix} \mathbf{f} = \mathbf{0}, \tag{2.1}$$

where $\mathbf{f} = (f_{11}, f_{21}, f_{31}, f_{12}, f_{22}, f_{32}, f_{13}, f_{23}, f_{33})^T$ denotes a vector obtained by rearranging the entries of the fundamental matrix in a column vector. Matrix $\mathbf{A}$ is made up of several sub matrices $\mathbf{W}_i$ of same dimension which is defined as,

$$\mathbf{W}_i = \begin{pmatrix} 2h_{11}^{\pi_i} & 0 & 0 & 2h_{21}^{\pi_i} & 0 & 0 & 2h_{31}^{\pi_i} & 0 & 0 \\ h_{12}^{\pi_i} & h_{11}^{\pi_i} & 0 & h_{22}^{\pi_i} & h_{21}^{\pi_i} & 0 & h_{32}^{\pi_i} & h_{31}^{\pi_i} & 0 \\ h_{13}^{\pi_i} & 0 & h_{11}^{\pi_i} & h_{23}^{\pi_i} & 0 & h_{21}^{\pi_i} & h_{33}^{\pi_i} & 0 & h_{31}^{\pi_i} \\ 0 & 2h_{12}^{\pi_i} & 0 & 0 & 2h_{22}^{\pi_i} & 0 & 0 & 2h_{32}^{\pi_i} & 0 \\ 0 & h_{13}^{\pi_i} & h_{12}^{\pi_i} & 0 & h_{23}^{\pi_i} & h_{22}^{\pi_i} & 0 & h_{33}^{\pi_i} & h_{32}^{\pi_i} \\ 0 & 0 & 2h_{13} & 0 & 0 & 2h_{23}^{\pi_i} & 0 & 0 & 2h_{33}^{\pi_i} \end{pmatrix}. \tag{2.2}$$

The entries of the matrix $\mathbf{W}_i$ originate from the homography $\mathbf{H}_i = \begin{pmatrix} h_{11}^{\pi_i} & h_{12}^{\pi_i} & h_{13}^{\pi_i} \\ h_{21}^{\pi_i} & h_{22}^{\pi_i} & h_{23}^{\pi_i} \\ h_{31}^{\pi_i} & h_{32}^{\pi_i} & h_{33}^{\pi_i} \end{pmatrix}$ which is induced by plane π_i. The least squares problem described in Eq. (2.1) is seriously ill-conditioned, which means that even under a tiny perturbation of any entry of matrix $\mathbf{A}$, the solution quickly diverges from the groundtruth result. Thus, the matrix $\mathbf{A}$ should be re-conditioned in order to stabilize its null space.

The presented method follows the idea of Hartley [1997] and introduces normalization in order to stabilize the result. However, it is not trivial to directly normalize the matrix $\mathbf{A}$ as it has been done in prior work for estimating the fundamental matrix or even the homography from point correspondences. The reason is two-fold. First, the normalization includes two parts, translation and scaling. The translation operation can only be performed by a linear transformation when the normalized object is described in the homogeneous form. Second, the normalization should be performed to data which have the same physical meaning.

The key to deal with the above two issues comes from the special structure of the matrix $\mathbf{F}^T\mathbf{H}$. The compatibility equation requires that $\mathbf{F}^T\mathbf{H}$ is a skew-symmetric matrix, and thus is of the form

$$\mathbf{F}^T\mathbf{H} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}. \tag{2.3}$$

The diagonal entries give three equations which describe an orthogonal relationship between corresponding column vectors of the fundamental matrix and a homography,

$$\mathbf{f}_i^T\mathbf{h}_i = 0, i = 1, 2, 3. \tag{2.4}$$

$\mathbf{f}_i$ and $\mathbf{h}_i$ denote the i_{th} column vector of the fundamental matrix $\mathbf{F} = \begin{pmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \mathbf{f}_3 \end{pmatrix}$ and the homography $\mathbf{H} = \begin{pmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{pmatrix}$. The other three equations enforce the skew symmetric property. However, only two of them are independent. This makes sense because a homography has 8 degrees of freedom (DoF). For the uncalibrated case, the intrinsic matrix is unknown which removes three constraints. Thus, only five independent constraints can be obtained from one homography, three from the orthogonal relationship described in Eq. (2.4) and the other two from the skew-symmetric property.

Our two-step reconditioning method realizes the non-trivial normalization by fully using the special structure of matrix $\mathbf{F}^T\mathbf{H}$. First, by utilizing the orthogonal relationship, we decompose the original least squares problem $\mathbf{A}\mathbf{f} = \mathbf{0}$ into three sub least squares problems $\mathbf{A}_i\mathbf{f}_i = \mathbf{0}$, where matrix $\mathbf{A}_i = \begin{pmatrix} \mathbf{h}_i^{\pi_1} & \mathbf{h}_i^{\pi_2} & \cdots & \mathbf{h}_i^{\pi_n} \end{pmatrix}^T$ and i $= 1, 2, 3$. Each column of the fundamental matrix $\mathbf{f}_i$ is estimated individually. The relative scale factor for each estimated solution $\mathbf{f}_i$ can then be recovered by using the skew-symmetric property of matrix $\mathbf{F}^T\mathbf{H}$ in Eq. (2.3). With this formulation, every column of matrix $\mathbf{A}_i$ has the same physical meaning. Besides, in order to perform the translation, the matrix $\mathbf{A}_i$ should be extended by an additional column $\mathbf{1}_{3\times1} = (1 \quad 1 \quad 1)^T$ which leads to $\tilde{\mathbf{A}}_i = [\mathbf{A}_i|\mathbf{1}_{3\times1}]$. Accordingly, the extended solution vector $\tilde{\mathbf{f}}_i$ is defined as $\tilde{\mathbf{f}}_i = \begin{pmatrix} \lambda_i^{-1}\mathbf{f}_i \\ 0 \end{pmatrix}$, where λ denotes the relative scale factor of the individually estimated solution. This extension turns each row of matrix $\mathbf{A}_i$ into the homogeneous form.

The mathematical proof is given after the whole algorithm is introduced.

The normalization is then performed by inserting a 4×4 linear transformation matrix $\mathbf{Q}_i$ and its inverse in between $\tilde{\mathbf{A}}_i$ and $\tilde{\mathbf{f}}_i$, resulting in

$$\tilde{\mathbf{A}}_i \mathbf{Q}_i \mathbf{Q}_i^{-1} \tilde{\mathbf{f}}_i = \hat{\mathbf{A}}_i \hat{\mathbf{f}}_i = \mathbf{0}, \tag{2.5}$$

where $\hat{\mathbf{A}}_i = \tilde{\mathbf{A}}_i \mathbf{Q_i}$ and $\hat{\mathbf{f}}_i = \mathbf{Q_i}^{-1} \tilde{\mathbf{f}}_i$. The linear transformation $\mathbf{Q}_i$ includes a translation and a scaling. We regard each $\mathbf{h}_i^{\pi_j}$ as a 3D point. Following the idea of Hartley [1997], the coordinates are translated such that the centroid $\mathbf{c}$ of the set of all such points becomes the origin. The coordinates are then scaled by applying an isotropic scaling factor s to all three coordinates of each point. Finally, we choose to scale the coordinates such that the average distance of a point $\mathbf{h}_i^{\pi_j}$ from the origin is equal to$\sqrt{3}$. The linear transformation $\mathbf{Q}_i$ and scaling related variables are defined as,

$$\mathbf{Q}_i = \begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ -c_1 s & -c_2 s & -c_3 s & 1 \end{pmatrix}, \tag{2.6}$$

$$\mathbf{c} = \begin{pmatrix} c_1 & c_2 & c_3 \end{pmatrix}^T = \frac{\sum_{j=1}^{m} \mathbf{h}_i^{\pi_j}}{m}, \tag{2.7}$$

$$s = \frac{\sqrt{3}}{\bar{d}}, \bar{d} = \frac{\sum_{j=1}^{m} \| \mathbf{h}_i^{\pi_j} - c \|_F}{m}. \tag{2.8}$$

The solution of the three sub least squares problems $\hat{\mathbf{A}}_i \hat{\mathbf{f}}_i = \mathbf{0}$ can be easily obtained via SVD. Then $\tilde{\mathbf{f}}_i = \mathbf{Q}_i \hat{\mathbf{f}}_i$. The only remaining task is to find the scale factors λ_i.

The skew-symmetric property of matrix $\mathbf{F}^T\mathbf{H}$ can be translated into another least squares problem $\mathbf{A}_\lambda \boldsymbol{\lambda} = \mathbf{0}$ via DLT, where $\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix}^T$ and $\mathbf{A}_\lambda$ is given by

$$\mathbf{A}_\lambda = \begin{pmatrix} \tilde{\mathbf{f}}_{1,1:3}^T \mathbf{h}_2^{\pi_1} & \tilde{\mathbf{f}}_{2,1:3}^T \mathbf{h}_1^{\pi_1} & 0 \\ \tilde{\mathbf{f}}_{1,1:3}^T \mathbf{h}_3^{\pi_1} & 0 & \tilde{\mathbf{f}}_{3,1:3}^T \mathbf{h}_1^{\pi_1} \\ 0 & \tilde{\mathbf{f}}_{2,1:3}^T \mathbf{h}_3^{\pi_1} & \tilde{\mathbf{f}}_{3,1:3}^T \mathbf{h}_2^{\pi_1} \\ \vdots & \vdots & \vdots \\ \tilde{\mathbf{f}}_{1,1:3}^T \mathbf{h}_2^{\pi_m} & \tilde{\mathbf{f}}_{2,1:3}^T \mathbf{h}_1^{\pi_m} & 0 \\ \tilde{\mathbf{f}}_{1,1:3}^T \mathbf{h}_3^{\pi_m} & 0 & \tilde{\mathbf{f}}_{3,1:3}^T \mathbf{h}_1^{\pi_m} \\ 0 & \tilde{\mathbf{f}}_{2,1:3}^T \mathbf{h}_3^{\pi_m} & \tilde{\mathbf{f}}_{3,1:3}^T \mathbf{h}_2^{\pi_m} \end{pmatrix}. \tag{2.9}$$

$\tilde{\mathbf{f}}_{i,1:3}$ in $\mathbf{A}_\lambda$ is defined as the first three rows of vector $\tilde{\mathbf{f}}_i$. $\mathbf{h}_i^{\pi_j}$ is defined the same as before. The full two-step linear method (TSL) is described in Algorithm 1.

It should be noted that in order to apply the normalization, the original least squares problem is modified. However, we will see in the following that solving the modified problem $\tilde{\mathbf{A}}_i \tilde{\mathbf{f}}_i = 0$ is equivalent to solving the original problem $\mathbf{A}_i \mathbf{f}_i = 0$. Therefore, two questions need to be answered in order to prove this claim:

1. After extending the matrix $\mathbf{A}_i$ by an additional column $\mathbf{1}_{3\times 1} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T$, what is the null-space configuration of $\tilde{\mathbf{A}}_i$?

Algorithm 1 Two-Step Linear Method (TSL)

1: Input: A collection of independently estimated homographies $\mathbf{H}_s$
2: **for** i = 1:3 **do**
3: $\quad \tilde{\mathbf{A}}_i = [\mathbf{A}_i | \mathbf{1}_{3\times 1}]$
4: $\quad \hat{\mathbf{A}}_i \leftarrow \tilde{\mathbf{A}}_i \mathbf{Q}_i$
5: $\quad \hat{\mathbf{f}}_i \leftarrow solve \hat{\mathbf{A}}_i \hat{\mathbf{f}}_i = \mathbf{0}$
6: $\quad \tilde{\mathbf{f}}_i \leftarrow \mathbf{Q}_i \hat{\mathbf{f}}_i$
7: **end for**
8: $\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix}^T \leftarrow solve \mathbf{A}_\lambda \lambda = \mathbf{0}$
9: Output: $\mathbf{F} = \begin{pmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \mathbf{f}_3 \end{pmatrix}$

2. Why does the solution of problem $\tilde{\mathbf{A}}_i \tilde{\mathbf{f}}_i = 0$ have the structure as $\tilde{\mathbf{f}}_i = \begin{pmatrix} \lambda_i^{-1} \mathbf{f}_i \\ 0 \end{pmatrix}$?

The answers are given by proving the following two claims:

- **Property 1**

  $Rank(\mathbf{A}_i) = 2, 1 \leq dim(N(\tilde{\mathbf{A}}_i)) \leq 2$ when the number of planes $m \geq 3$, , where $\mathbf{N}(\cdot)$ denotes the null space of $(\cdot)$

- **Property 2**

  $\mathbf{N}(\tilde{\mathbf{A}}_i) = \begin{pmatrix} \mathbf{N}(\mathbf{A}_i) \\ 0 \end{pmatrix}$.

Proof. **Property 1**

Assuming that two camera matrices are given by $\mathbf{P}_1 = [\mathbf{I}_{3\times 3} | \mathbf{0}_{3\times 1}]$ and $\mathbf{P}_2 = [\mathbf{B}|\mathbf{b}]$, each homography induced by a plane $\boldsymbol{\pi}_j = [-\mathbf{v}_j^T, 1]$ observed by the two cameras can be denoted as

$$\mathbf{H}_j^{\pi} \simeq \mathbf{B} + \mathbf{b}\mathbf{v}_j^T. \tag{2.10}$$

Each row of matrix $\mathbf{A}_i$ contains the i_{th} column of one homography, which gives

$$\mathbf{h}_i^{\pi_j} \simeq \mathbf{B}_i + \mathbf{v}_{j,i}\mathbf{b}, \tag{2.11}$$

where $\mathbf{B}_i$ denotes the i_{th} column of the matrix $\mathbf{B}$ and $\mathbf{v}_{j,i}$ the $\mathbf{i}_{th}$ element of the vector $\mathbf{v}_j$. It is obvious to see that if we regard each row of matrix $\mathbf{A}_i$ as a general 3D point, all the points $\mathbf{h}_i^{\pi_j}$ are lying on the line with the direction of $\mathbf{v}_{j,i}\mathbf{b}$ passing point $\mathbf{B}_i$. Thus $Rank(\mathbf{A}_i) = 2$.

Since matrix $\tilde{\mathbf{A}}_i$ is obtained by adding an additional column $\mathbf{1}_{3\times 1}$ to $\mathbf{A}_i$, it is also obvious to see that

$$Rank(\mathbf{A}_i) \leq Rank(\tilde{\mathbf{A}}_i) \leq 3. \tag{2.12}$$

Because

$$Rank(\tilde{\mathbf{A}}_i) + dim(N(\tilde{\mathbf{A}}_i)) = 4, \tag{2.13}$$

thus we finally have

$$1 \leq dim(N(\tilde{\mathbf{A}}_i)) \leq 2.^1 \tag{2.14}$$

[1] If $Rank(\tilde{\mathbf{A}}_i) = 3$, $\tilde{\mathbf{A}}_i$ has only a one dimensional null space which is the eigen vector corresponding to the

□

Proof. **Property 2**

Assuming $\mathbf{x} \in \mathbf{N}(\mathbf{A}_i)$, *and* $\tilde{\mathbf{x}} \in \mathbf{N}(\tilde{\mathbf{A}}_i)$, we have $\mathbf{A}_i\mathbf{x} = \mathbf{0}$, *and* $\tilde{\mathbf{A}}_i\tilde{\mathbf{x}} = \mathbf{0}$.

Obviously, $\forall \mathbf{x} \in \mathbf{N}(\mathbf{A}_i)$, $\tilde{\mathbf{A}}_i \begin{pmatrix} \mathbf{x} \\ 0 \end{pmatrix} = [\mathbf{A}_i | \mathbf{1}_{3\times 1}] \begin{pmatrix} \mathbf{x} \\ 0 \end{pmatrix} = \mathbf{0}$.

Thus, $\begin{pmatrix} \mathbf{N}(\mathbf{A}_i) \\ 0 \end{pmatrix} \in \mathbf{N}(\tilde{\mathbf{A}}_i)$.

Necessary condition Q.E.D.

On the other hand, assume $\forall \tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ \omega \end{pmatrix}, \omega \neq 0$.

Since $\tilde{\mathbf{A}}_i\tilde{\mathbf{x}} = \mathbf{0}$,

$\Rightarrow \mathbf{A}_i\mathbf{x} + \omega\mathbf{1}_{3\times 1} = \mathbf{0}$,

$\Rightarrow \omega\mathbf{1}_{3\times 1} = \mathbf{0}$,

$\Rightarrow \omega = \mathbf{0}$,

$\Rightarrow$ contradiction,

Thus, $\mathbf{N}(\tilde{\mathbf{A}}_i) \in \begin{pmatrix} \mathbf{N}(\mathbf{A}_i) \\ 0 \end{pmatrix}$.

Sufficient condition Q.E.D.

Summarizing, $\mathbf{N}(\tilde{\mathbf{A}}_i) = \begin{pmatrix} \mathbf{N}(\mathbf{A}_i) \\ 0 \end{pmatrix}$. □

The mathematical proofs above explain why we can get the solution to the original problem by solving the reconditioned least squares problems. One drawback of the proposed method is that at least 3 planes (homographies) are needed for computing the fundamental matrix. The reason lies in the normalization. The matrix $\mathbf{A}_i$ is extended by an additional column $\mathbf{1}_{1\times 3} = (1 \quad 1 \quad 1)^T$. Thus, with only two homographies, the rank of $\tilde{\mathbf{A}}_i$ is always 2, and the normalization cannot be applied.

2.3 Experiment

In this section, we compare the performances of DLT, HP and TSL on both synthetic and real data. Numerical stability of DLT and TSL as well as algorithmic complexity of the three methods are also discussed.

The input homographies can be derived from either point or line features as they are dual geometric entities (Guerrero and Sagues [2001]; Guerrero and Sagüés [2003]; Dubrofsky [2009]). We use line features during the synthetic experiments, and point correspondences during the experiment on real data.

smallest eigenvalue of matrix $\tilde{\mathbf{A}}_i$. Otherwise, if $Rank(\tilde{\mathbf{A}}_i) = 2$, the final solution of problem $\tilde{\mathbf{A}}_i\tilde{\mathbf{f}}_i$ resides in a two dimensional null space. However, during our experiment, we never observed the case of $Rank(\tilde{\mathbf{A}}_i) = 2$.

2.3.1 Synthetic Experiment

For each single experiment, we construct two artificial views observing planes in a 3D environment. Groundtruth motion and structure (planes) is generated in the same way as in (Szpak et al. [2014]). Without loss of generality, the camera pose of the first view is assumed to be identical with the world frame. The absolute pose of the second view is defined by motion parameters lying within a certain range. The rotation angles along each axis (roll, pitch, yaw) lie within $(-5^\circ, 5^\circ)$ and the translation in each direction (*X, Y, Z*) is within $(-100, 100)$. The structure is randomly generated by creating $N = 5$ planes with known homographies. Four groups of Gaussian noise ($\mu = 0, \sigma \in [0, 0.5]$) contaminated points[2] are created on each plane, which are used for fitting the line features. The image size is 640×480 and the focal length is $f_x = f_y = 250$. The relative motion parameters are extracted from the estimated fundamental matrix $\mathbf{F}$ (in fact from essential matrix $\mathbf{E} = \mathbf{K}^T\mathbf{F}\mathbf{K}$).

As shown in Fig 2.1, both HP and TSL outperform DLT in the accuracy of the estimated fundamental matrix and the motion parameters. We use max norm of the difference between $\mathbf{F}_{\text{groundtruth}}$ and $\mathbf{F}_{\text{estimated}}$ as a criterion for assessing the accuracy of the estimated $\mathbf{F}$. The estimated rotation matrix is compared to the groundtruth by computing the angle $\Theta = \arccos(\frac{\text{trace}(\mathbf{R}^T_{\text{groundtruth}}\mathbf{R}_{\text{estimated}})-1}{2})$. The estimated translation is compared against the groundtruth by computing the angle between two translation vectors $\mathbf{t}_{\text{groundtruth}}$ and $\mathbf{t}_{\text{estimated}}$. TSL is more noise resilient in terms of the fundamental matrix estimation in comparison to HP. Concerning the accuracy of the extracted motion parameters, TSL and HP perform equally well.

2.3.2 Experiment on Real Images

The algorithm is tested on the famous Oxford Corridor sequence. Homographies are estimated from Harris corner correspondences (Harris and Stephens [1988]). Points on each plane are grouped manually and outliers are rejected by applying the Random sample consensus (RANSAC) technique (Fischler and Bolles [1981]).

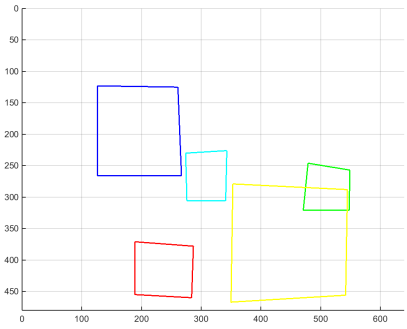
As shown in Fig. 2.2, the epipole estimated by TSL (i.e. the intersection point of blue lines) is closest to the groundtuth. The epipole $\mathbf{e}$ is extracted from the null space of the fundamental matrix. A small error in any entry of the fundamental matrix can easily cause the resulting epipole to severly deviate from the groundtruth location.

We can easily see that our conclusions from the synthetic experiment are verified, namely that the proposed method clearly outperforms DLT and shows advantages over HP as well.

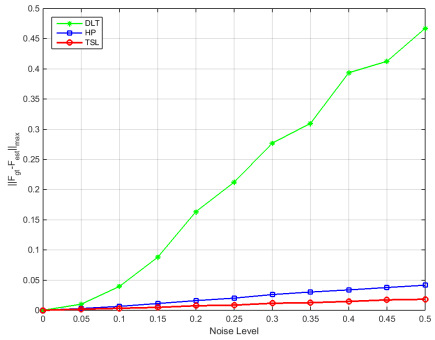
2.3.3 Numerical Stability and Algorithmic Complexity

It is easy to understand why the performance of DLT can be dramatically improved by including normalization. Without the normalization, as shown in Eq. (2.1) and Eq. (2.2), some of the entries are smaller than the others by several orders of magnitude which directly causes the

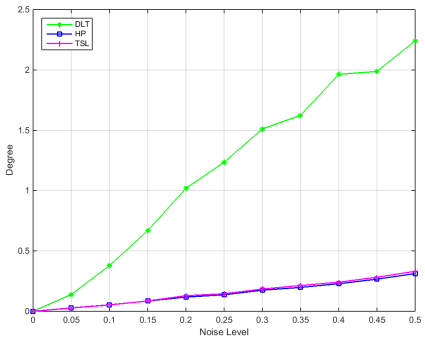
[2] As shown in (Zeng et al. [2008]), when the line is close to or passing through the origin of the coordinate frame, the quality of the estimated homographies decreases dramatically. This problem can be solved by performing a prior normalization to the line parameters. For the sake of simplicity and without losing generality, the lines generated in our experiment are forced to be away from the origin of the coordinate frame by at least 10 pixels.



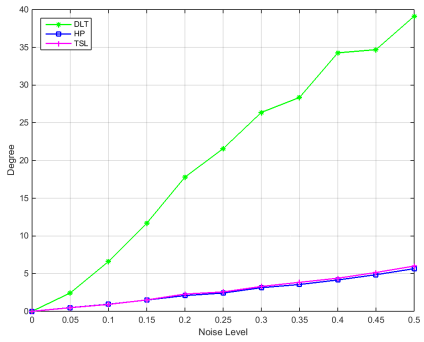
(a) Synthetic experiment configuration.



(b) Max norm of difference between groundtruth and estimated F.



(c) Error in rotation.



(d) Error in translation.

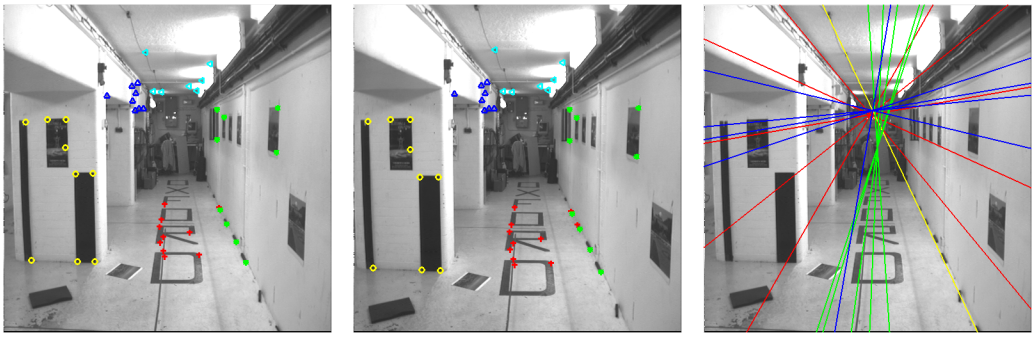
Figure 2.1: Figure (a) shows the configuration of the experiment. The accuracy of the fundamental matrix estimation is shown in Fig (b) with the max norm as the assessing criterion. Figures (c) (d) separately depict rotation and translation error of DLT, HP, TSL.

serious ill-conditioning of the original least squares problem. We record the numerical stability of DLT and TSL. As can be seen in Fig 2.3, the condition numbers of the three normalized sub least squares problem are far smaller than the one of the DLT solution. The average variance of the condition number also demonstrates that TSL is numerically more stable. A simple complexity comparison is given in Tab. 2.1. In our experiment, $N = 80$ and $M = 5$. TSL and HP lead to similar performances under these conditions, while TSL needs less computational resources than HP does.

Table 2.1: Algorithm Complexity Comparison

Method	Input	Matrix size to be solved
HP	N points (not coplanar, $N \geqslant 8$)	$\mathbf{A}_{N\times 9}$
TSL	M planes ($M \geqslant 3$)	$3 \times \mathbf{A}_{M\times 4} + \mathbf{A}_{3M\times 3}$

It is worth pointing out that, during the experiment, we discovered that if the consistency among the inter-frame homographies is guaranteed, the estimated fundamental matrix is al-



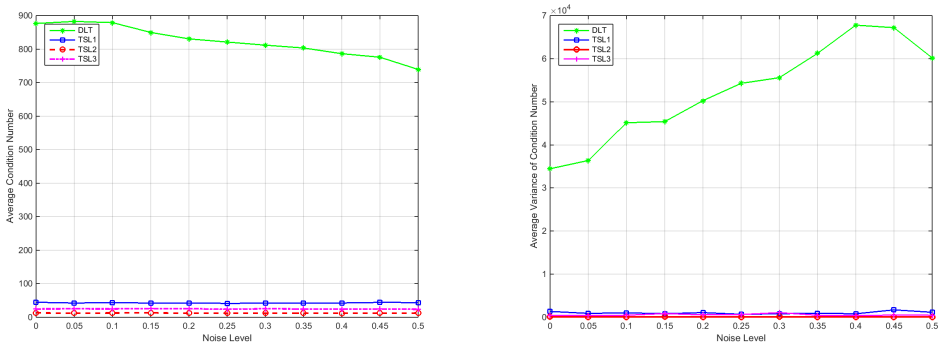
(a) Grouped point features in image one

(b) Grouped point features in image two

(c) Epipolar lines of groundtruth and all three methods

Figure 2.2: Grouped point features which are used for estimating the homographies are shown in Fig (a) and (b). Epipolar lines obtained by DLT(yellow), HP(green), TSL(blue) and groundtruth (red) are shown in Fig (c).

ways accurate and robust no matter which method is used. Typically, perfect consistency constraints are available only in an implicit form, which can only be achieved by iterative non-linear methods, e.g. Joint Bundle Adjustment (BA-Joint) and AML (Szpak et al. [2014]; Chojnacki et al. [2015]). Explicit methods like (Shashua and Avidan [1996]; Zeinik-Manor and Irani [2002]; Chen and Suter [2009]) use a low-rank approximation under the Frobenius norm or the Mahalanobis norm to enforce the rank-four constraint. However, the explicit form is derived from a relaxed consistency constraint which means the consistency cannot be perfectly guaranteed. This discovery in fact gives an alternative explanation to why the direct estimation of the fundamental matrix by the compatibility equation is not stable.



(a) Average condition number of DLT and TSL

(b) Average variance of condition number

Figure 2.3: The average condition number under each noise level is shown in Figure (a). TSL1, TSL2 and TSL3 are the three sub least squares problems of TSL. Figure (b) shows the corresponding average variance of the condition number.

2.4 Conclusion

This chapter revisits an old topic: accurately and robustly estimating the fundamental matrix given a collection of independently estimated homograhies. We first review three classical methods and then show that a simple but non-trivial two-step normalization within the direct linear method achieves similar performance than the more computationally intensive hallucinated points based method. We verify the correctness and robustness of our method by giving a mathematical proof and an experimental evaluation on both synthetic and real data. The numerical stability analysis and algorithm complexity discussion finally demonstrates our improvement and further advantages of the proposed technique.

Chapter 3

Real-time Rotation Estimation for Depth Sensors in Piece-wise Planar Environments

In the previous chapter, we have seen how to solve epipolar geometry between two perspectives captured by an uncalibrated camera. In this chapter, we look into more practical solutions and start our investigation of efficient algorithms for RGB-D cameras, and in particular look at the case of an RGB-D camera exploring a piece-wise planar environment.

3D depth sensors such as RGB-D cameras are a popular alternative to classical cameras for the purpose of autonomous navigation and robotic perception. Active sensors are particularly advantageous when it comes to structures with homogeneously colored surfaces, textureless environments, or even operation in darkness. The point clouds produced by these sensors come in metric scale. They can be used directly to perform point registration via the iterative closest point method (ICP) (Besl and McKay [1992]), thus resulting in motion estimation in absolute scale. However, ICP-based motion estimation is either too easy to get trapped in local minima, or too computationally expensive to meet the requirements of real-time applications. Considering the fact that rotational drift is an important part of the inaccuracy of position estimation, the goal of this chapter is to develop an efficient and piece-wise drift-free 3D rotation estimation method for RGB-D cameras operating in man-made environments.

Our approach relies on surface normal vectors, which can be extracted directly from point clouds, and convey rich geometric information for applications like scene segmentation and object classification (Wei et al. [2014]), structure and pose estimation (Glover et al. [2012]; Schwarz et al. [2015]), and even grasping or manipulation (Stückler et al. [2011]). Normal vector distributions typically contain a special structure due to the vast availability of planar surfaces in man-made environments. These structural regularities notably lead to modes in the normal vector density distribution.

Rotation estimation for depth cameras by exploiting the organized structure of surface normal vector distributions has been studied previously. However, existing works are limited to either strict Manhattan World (MW) environments (Coughlan and Yuille [1999]; Straub et al. [2015b]) or the further relaxed Mixture of Manhattan Frames (MMF) case (Straub et al. [2014]). Following the idea of (Straub et al. [2014, 2015b]), we also exploit surface normal vector distributions, but extend it to the more general case of piece-wise planar environments

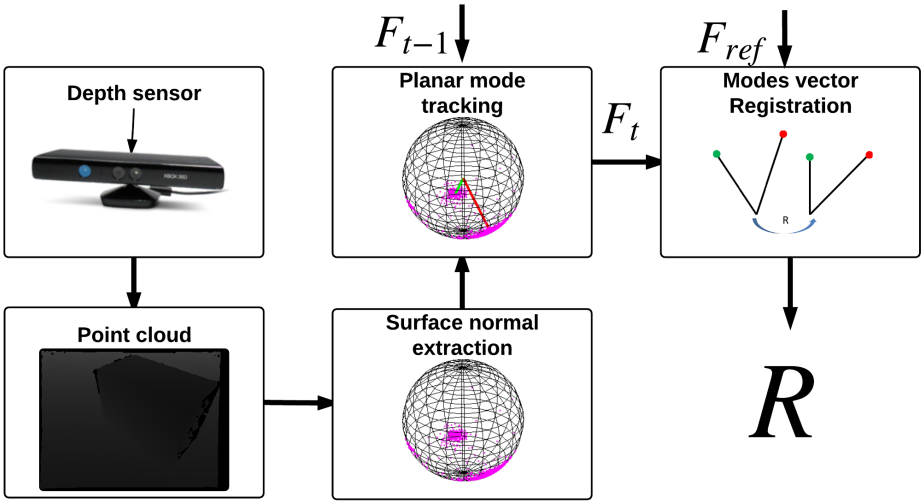


Figure 3.1: Overview of the proposed 3D rotation estimation algorithm for depth cameras in piece-wise planar environments.

with arbitrary pieces of slanted planes.

The contribution of this chapter is three-fold:

- Assuming that there are several dominant planes in the environment, we present a non-parametric method for discovering and tracking planar modes in the density distribution of the surface normal vectors. It is a mean-shift algorithm that operates on the unit sphere, and avoids the need of estimating the parameters of a complete explicit model of the density distribution function.

- Second, we present a robust and piece-wise drift free rotation estimation method which solves the joint registration of pairs of corresponding planar modes in a general ℓ_1-norm regression scheme. This algorithm works robustly with up to 50% of badly tracked modes.

- We introduce a basic memory scheme that remembers dying planar modes. We show that the memory is capable of further compensating drift when previously visited planar structures are reobserved. This functionality has similarities with loop closures in classical SLAM.

The result is a simple but accurate, robust and highly efficient strategy for online tracking of the rotation of a depth camera. The remaining part is organized as follows: Related work on this topic are reviewed in Section 3.1. Section 3.2 declares all mathematical notations used in this chapter as well as all underlying assumptions. Section 3.3 presents the core of our method. Section 3.4 finally gives a performance and robustness analysis on both synthetic and real datasets. We conclude with a summary and a discussion.

3.1 Related Work

Online rotation estimation is related to odometry or motion estimation in general. We limit the discussion to solutions that utilize active sensors such as LIDARs and RGB-D cameras because we only use depth information in this work. The most commonly used method is given by the ICP algorithm (Besl and McKay [1992]) which performs registration through iterative minimization of the sum of squared distances between spatial neighbors in two point clouds. Classical ICP based methods are prone to local minima as soon as the displacement increases and thus the point cloud structure is subjected to intensive changes. In order to tackle situations of large view-point changes, the community has therefore investigated globally optimal solutions to the point set registration problem, such as (Yang et al. [2013]). These methods are however inefficient and thus not suited for real-time applications on CPU. Even the most recent local ICP methods (Pomerleau et al. [2011, 2013]) achieve real-time frame rate for the QVGA resolution only (e.g. 320×240 pixel).

The upcoming of RGB-D cameras has however led to a new generation of 2D-3D registration algorithms that exercise a hybrid use of both depth and RGB information. Kerl et al. [2013b] for instance uses the depth information along with the optimized relative transformation to warp the image from one frame to the next, thus permitting direct and dense photometric error minimization. Our algorithm is evaluted on datasets captured by a Microsoft Kinect. A comparison of our results to the method presented in (Kerl et al. [2013b]) is also provided.

There are some recent works that directly build on top of surface normal vectors. By exploiting the structural regularity of man-made environments, Straub et al. [2015b] present a real-time maximum a posteriori (MAP) inference of the local Manhattan Frame (MF). This work heavily relies on GPU resources for a real-time inference of a parametric model, and is furthermore strictly limited to the Manhattan world scenario. More general, non-parametric model estimation is presented in (Straub et al. [2014]), which can handle the arbitrary piecewise planar case. While strongly related to our work, the method in (Straub et al. [2014]) is more computationally expensive and aims at scene understanding and segmentation rather than accurate rotation estimation.

3.2 Problem Definition and Prerequisites

Our main assumption is that the environment is static and consists of multiple pieces of planar structures. Under this assumption, the surface normal vectors $\mathbf{N}^{\mathbf{C}} = [\mathbf{n}_1, \ldots, \mathbf{n}_M]$ distribute in an organized and distinctive manner on the unit sphere[1]. Given surface normal vectors extracted from point clouds by using the method in (Holz et al. [2012]), our goal is two-fold:

- Discover and keep track of the planar modes $\mathbf{F} := [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N]$ on the unit sphere. $\mathbf{F}$ is a $3 \times N$ matrix which defines a bundle of planar direction vectors $\mathbf{f}_i$. For simplicity, we call $\mathbf{F}$ a *bundle*.

- Estimate the relative rotation $\mathbf{R}$ between the reference and the current frame such that $\mathbf{F}^{cur} \simeq \mathbf{R}\mathbf{F}^{ref}$. $\simeq$ means that the equality is valid up to noise or outliers.

[1] Superscript $^{\mathbf{C}}$ denotes that the surface normal vectors are described in the coordinate system of the sensor.

By a reference frame, we refer to a frame that is

- the very first frame in the sequence where planar modes are initially discovered.
- a frame further in the sequence selected upon a *bundle update*. During tracking, existing modes may die or new modes may be discovered which leads to a so-called *bundle update*.

3.3 Normal-vector based Rotation Estimation

The surface normal vectors $\mathbf{N}^{\mathbf{C}}$ of piece-wise planar structures always have some organized distribution on the unit sphere $\mathbb{S}^2$ which can be exploited to track the orientation of the depth camera. It is reasonable to assume that these unit vectors $\mathbf{n}_i$ are samples of a probability density function, as they are more likely to be distributed around the normal vectors of the plane pieces. The process of finding these planar direction vectors is therefore equivalent to mode-seeking in this density distribution (i.e. finding the local maximum in the density distribution function).

A popular, fast, and notably non-parametric method to seek modes is given by the mean shift algorithm (Carreira-Perpiñán [2015]). Given an approximate location for a mode, the algorithm applies local Kernel Density Estimation (KDE) to iteratively take steps in the direction of increasing density. We apply this idea to our unit normal vectors on the manifold $\mathbb{S}^2$ using a Gaussian kernel over conic section windows of the unit sphere. The result is optimal under the assumption that the angles between the normal vectors and their corresponding mode centres have a Gaussian distribution. We track the bundle by simply tracking each individual mode independently. Each mode is tracked by starting from its previous position on the unit sphere. While this means that we allow inter-mode angle variation during the tracking of the bundle $\mathbf{F}^{cur}$, we follow the mode-tracking by registering the entire bundle with respect to a fixed bundle $\mathbf{F}^{ref}$ in a reference frame, thus avoiding drift-effects.

3.3.1 Mean-shift on the Unit Sphere

The core of our method is a single mean shift iteration for each planar mode given a set of normal vectors on $\mathbb{S}^2$. It works as follows:

- We start by finding all normal vectors that are within a neighbourhood of the considered centre $\mathbf{f}_j$. The range of this neighbourhood is notably defined by the width of our kernel for the KDE. In our case, the window is a conic section of the unit sphere and the apex angle of the cone θ_{window} defines the size of the local window. Relevant normal vectors $\mathbf{n}_i$ for mode j need to lie inside the respective cone, and thus pass the condition

$$\angle(\mathbf{n}_i, \mathbf{f}_j) < \frac{\theta_{\text{window}}}{2}. \tag{3.1}$$

  Let us define the index i_j which iterates through all $\mathbf{n}_i$ that fulfill the above condition.

- We then project all contributing $\mathbf{n}_{i_j}$ into the tangential plane at $\mathbf{f}_j$ in order to compute a mean shift. Let $\mathbf{Q}$ represent the rotation matrix that rotates $\mathbf{f}_j$ to $[0,0,1]^T$. $\mathbf{Q}$ can be

obtained by

$$\mathbf{Q} = \mathbf{I} + [\mathbf{v}]_\times + [\mathbf{v}]_\times^2 \frac{1-c}{s^2}, \tag{3.2}$$

where $\mathbf{v} = \mathbf{f}_j \times [0,0,1]^T$, $s = \|\mathbf{v}\|$, $c = \mathbf{f}_j^T [0,0,1]^T$, and $[\mathbf{v}]_\times$ is the skew-symmetric matrix of $\mathbf{v}$. Then

$$\mathbf{n'}_{i_j} = \mathbf{Q}\mathbf{n}_{i_j} \tag{3.3}$$

represents the normal vectors rotated such that the last coordinate is along the direction of $\mathbf{f}_j$. In order for the distances in the tangential plane to represent proper geodesics on $\mathbb{S}^2$ (or equivalently angular deviations), we apply the Riemann exponential map. The rescaled coordinates in the tangential plane are given by

$$\begin{aligned} \mathbf{m'}_{i_j} &= \frac{\sin^{-1}(\lambda)\,\mathrm{sign}(n'_{i_j,z})}{\lambda} \begin{bmatrix} n'_{i_j,x} \\ n'_{i_j,y} \end{bmatrix}, \\ \text{where } \lambda &= \sqrt{n'^2_{i_j,x} + n'^2_{i_j,y}}. \end{aligned} \tag{3.4}$$

Note that—due to the factor $\mathrm{sign}(n'_{i_j,z})$—this projection has the advantage of correctly projecting normal vectors from either direction sense into the same tangential plane.

- We compute the mean shift in the tangential plane

$$\mathbf{s'}_j = \frac{\sum_{i_j} e^{-c\|\mathbf{m'}_{i_j}\|^2} \mathbf{m'}_{i_j}}{\sum_{i_j} e^{-c\|\mathbf{m'}_{i_j}\|^2}}. \tag{3.5}$$

  c is a design parameter that defines the width of the kernel in the tangential plane. It can be derived from θ_{window}.

- To conclude, we transform the mean shift back onto the unit sphere using the Riemann logarithmic map. The update mode $\mathbf{f}_j^\star$ is finally obtained by compensating the rotation $\mathbf{Q}$.

$$\mathbf{f}_j^\star = \mathbf{Q}^T \overline{\begin{bmatrix} \frac{\tan(\|\mathbf{s'}_j\|)}{\|\mathbf{s'}_j\|} \mathbf{s'}_j^T & 1 \end{bmatrix}}^T, \tag{3.6}$$

  where $\overline{[\cdot]}$ returns the input 3-vector divided by its norm.

3.3.2 Robust Rotation Estimation

Once the new location of each mode of the bundle $\mathbf{F}$ has been tracked, the rotation from the reference frame to the current frame can be obtained by applying a least-squares fitting method (Arun et al. [1987]). Each mode of $\mathbf{F}^{ref}$ and $\mathbf{F}^{cur}$ is regarded as a 3D point. This reduces the

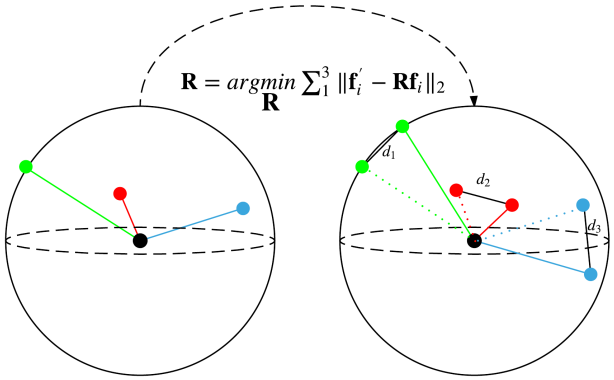


Figure 3.2: Illustration of the geometry of the problem. Three modes exist in both the reference view (left) and the current view (right). The chordal distance d_i between each corresponding pair of modes is indicated with a black line segment. The relative rotation from the reference view to the current view is the solution that minimizes the sum of the chordal distances (in a general sense of ℓ_1-norm regression).

problem to finding a rotation $\mathbf{R}$ that minimizes the cost function

$$\begin{aligned} \Sigma^2 &= \sum_{i=1}^{N} (\mathbf{f}_i^{cur} - \mathbf{R}\mathbf{f}_i^{ref})^T (\mathbf{f}_i^{cur} - \mathbf{R}\mathbf{f}_i^{ref}) \\ &= \sum_{i=1}^{N} (\mathbf{f}_i^{curT}\mathbf{f}_i^{cur} + \mathbf{f}_i^{refT}\mathbf{f}_i^{ref} - 2\mathbf{f}_i^{curT}\mathbf{R}\mathbf{f}_i^{ref}) \end{aligned} \tag{3.7}$$

This cost function has a geometric meaning as shown in Fig. 3.2. Each item of the cost function is the square of the chordal distance between a pair of corresponding modes on the unit sphere. Minimizing Σ^2 therefore is equivalent to finding the closest bundle near $\mathbf{F}^{cur}$ that has same inter-mode angles than $\mathbf{F}^{ref}$, and notably under an ℓ_2-metric (i.e. squared chordal distances).

We apply Arun's method (Arun et al. [1987]). Minimizing Σ^2 is equivalent to maximizing the third cost term because the previous terms are constant. The original minimization problem therefore turns into maximizing

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{N} \mathbf{f}_i^{curT}\mathbf{R}\mathbf{f}_i^{ref} \\ &= Trace(\sum_{i=1}^{N} \mathbf{R}\mathbf{f}_i^{ref}\mathbf{f}_i^{curT}) = Trace(\mathbf{R}\mathbf{H}) \end{aligned} \tag{3.8}$$

where $\mathbf{H} := \sum_{i=1}^{N} \mathbf{f}_i^{ref}\mathbf{f}_i^{curT}$. Let the SVD of $\mathbf{H}$ be $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$. The best rotation matrix is $\mathbf{R} = \mathbf{V}\mathbf{U}^T$. A reflection check is necessary for the case of $det(\mathbf{R}) = -1$. A detailed mathematical proof can be found in (Arun et al. [1987]).

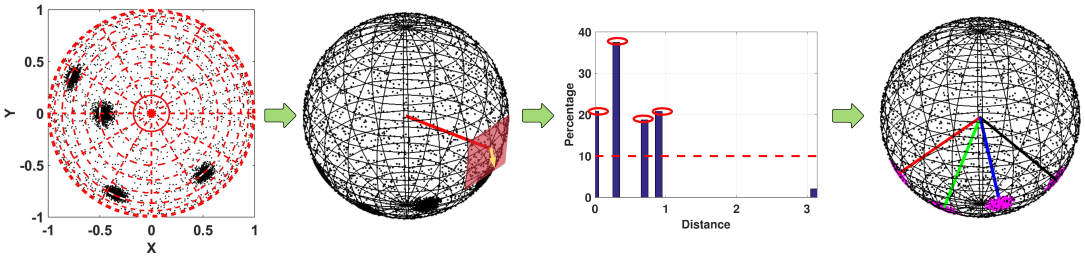


Figure 3.3: Initialial mode seeking. The first figure shows the pattern that defines the starting coordinates for the mean-shift clustering. The second figure shows a mean-shift in a tangential plane starting from a given coordinate. The histogram-based non-maximum suppression is shown in the third figure. It splits off mode centres by picking one mode and creating a histogram of rotation distances with respect to all other modes. The final result after non-maximum suppression is shown in the last figure. Four planar modes are found and highlighted with different colors.

For the sake of robustness, we replace the least-squares method with a robust general ℓ_1-norm regression scheme. The new optimization problem becomes

$$\mathbf{R} = \arg\min_{\mathbf{R}} \sum_{i=1}^{n} |\mathbf{f}_i^{cur} - \mathbf{R}\mathbf{f}_i^{ref}| \tag{3.9}$$

where $|\cdot|$ returns the length of a given vector. The most common tool for solving ℓ_p-norm regression problems with an objective function format like Eq. 3.9 is the iteratively reweighted least squares (IRLS) method (wikipedia.org). In our case, iterative reweighting is easily done by iteratively finding the rotation matrix $\mathbf{R}_k$ that maximizes

$$\begin{aligned} \mathcal{L} &= Trace(\sum_{i=1}^{N} w_i \mathbf{R}_k \mathbf{f}_i{}^{ref} \mathbf{f}_i^{curT}), \text{ where} \\ w_i &= |\mathbf{f}_i^{cur} - \mathbf{R}_{k-1}\mathbf{f}_i^{ref}|^{-1}. \end{aligned} \tag{3.10}$$

As this remains a linear problem in each iteration, Arun's method (Arun et al. [1987]) remains applicable. Section 3.4.2 illustrates the benefit of the ℓ_1-extension. The pseudo code of bundle tracking and robust rotation estimation is given in Alg. 2.

3.3.3 Initialization and Bundle Update

We use mean-shift clustering to initialize the algorithm, and thus build on top of our planar mode tracking scheme. The procedure is summarized in Fig. 3.3. In order to guarantee that the mode-seeking covers the whole space, the unit sphere is divided equally along longitudes and latitudes which gives a set of starting coordinates for the mean-shift tracking. Mean-shift iterations starting from neighboring coordinates may converge to the same mode, which is why we clean the identified set of modes by a histogram-based non-maximum suppression.

New modes may appear or disappear as the view-point changes. If the density of surface

Algorithm 2 Bundle tracking and rotation estimation.

function BundleTracking($\mathbf{N}^{\mathbf{C}}$,$\mathbf{F}^{ref}$,$\mathbf{F}^{t}$)
$\mathbf{F}^{t+1} = \emptyset$
for each $\mathbf{f}_i^t$ **do**
  if $\mathbf{f}_i^t$ is not dying **then**
    $\mathbf{f}_i^{t+1} \leftarrow$ Mean-shift based mode update.
    Push back $\mathbf{f}_i^{t+1}$ to $\mathbf{F}^{t+1}$
  end if
end for
if numel($\mathbf{F}^{t+1}$) < 2 **then**
  return []. ▷ Tracking lost.
end if
$w_i = 1, i = 1, 2, ..., N$ ▷ N = number of mode pairs.
while **R** does not converge **do**
  $\mathbf{H} = \sum_{i=1}^{N} w_i \mathbf{f}_i^{ref} {\mathbf{f}_i^{t+1}}^T$
  $\mathbf{U_R}\mathbf{\Sigma}_R\mathbf{V_R^t} \leftarrow$ svd($\mathbf{H}$)
  $\mathbf{R} = \mathbf{V_R}{\mathbf{U_R}}^T$ ▷ Validity Check, see (Arun et al. [1987]).
  $w_i = \frac{1}{\max(\delta, |\mathbf{f}^{t+1} - \mathbf{R}\mathbf{f}^{ref}|)}, i = 1, 2, ..., N$ ▷ δ is a small number
end while
if New born mode appears **then**
  Push back $\mathbf{f}^{\star}$ to $\mathbf{F}^{t+1}$
  Update $\mathbf{F}^{ref} \leftarrow \mathbf{F}^{t+1}$
end if
return **R**,$\mathbf{F}^{t+1}$,$\mathbf{F}^{ref}$.
end function

normal vectors in one mode decreases to less than a designed threshold, the mode is deemed dying and removed from the reference bundle $\mathbf{F}^{ref}$. We find new modes by a mode discovery module, and update the reference bundle $\mathbf{F}^{ref}$ each time a new mode is found[2]. The mode-discovery module continuously monitors the number of surface normal vectors in each cell of the above mentioned grid. If a new mode appears, the number of the surface normal vectors in that direction will grow substantially, thus triggering mean-shift tracking from the center of the cell. Note that this operation is much more expensive than simple mode tracking. We therefore run this monitoring in a separate thread and at a lower frame rate, thus maintaining real-time performance for the actual rotation estimation.

3.3.4 Memory Function

Instead of simply removing dying modes, we keep forecasting their directions in the current frame using the estimated rotation (even if no normal vectors are currently associated to it). We call the set of inactive modes a mode memory. If a new planar piece is discovered, and the new-born mode is close to an inactive mode in the memory, we reactivate this mode rather

[2]Note that—in order to reduce drift—we simply rotate persisting modes forward rather than replacing them by their tracked equivalent.

than replacing it with a new one. This association compensates drift since the mode became inactive (and notably about the axis that this mode corresponds to). We will see in Section 3.4.3 that this reduces long-term drift.

3.4 Experimental Evaluation

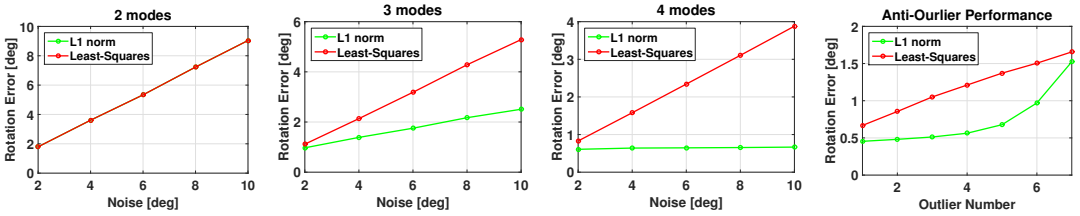
Now we proceed to the evaluation of the presented algorithm. The parameter values chosen in our experiments are first provided. Then a dedicated simulation experiment is presented showing the importance of the general ℓ_1-norm regression scheme towards the robustness of the rotation estimation. We also test the algorithm on a custom synthetic dataset which demonstrates the piece-wise drift-free property and long-term drift resilience with activated memory function. Finally we evaluate the proposed algorithm on a set of publicly available, real datasets and compare our results directly to another two state-of-the-art depth camera tracking solutions.

3.4.1 Parameter Configuration

The apex angle of the conic section corresponding to the width of the kernel for the mode tracking is set to 40° during initialization, and 20° during tracking. By using a larger apex angle in initialization, it is more likely that more seeking trials starting from different coordinates in a neighborhood would converge to the same local maximum which will be picked as a mode in the following. The reduction of the cone apex angle in tracking is justified by the assumption that the orientation of the bundle does not change too much under smooth motion. Each iterative mean-shift procedure terminates once the angle between two successive updates falls below a threshold angle θ_{converge}, which we set to 1°. The factor c in Eq. 4.5 is set to 20. Mean-shift updates are furthermore required to have a minimum number N_{min} of surface normal vectors within the conic window, which is set to 10% of the total number of surface normal vectors. N_{min} is also the threshold for checking dying modes.

3.4.2 Simulation Experiments

We provide a dedicated simulation to show that our algorithm can work robustly in a situation where some of the modes are badly tracked. The first part of this simulation consists of a series of three experiments during which we perform a registration of bundles with 2, 3, and 4 modes. In each experiment, all the modes are perturbed by Gaussian noise. In addition, an elevated amount of noise is added to one of the modes only, which simulates a situation in which the tracking of that particular mode fails. The case of disturbed surface normal vector measurements may happen for various reasons, including heavily inclined planar pieces, a reflection on a smooth surface, or a moving element in the scene. We each time compare the performance of our general ℓ_1-norm regression scheme to that of the original least-squares method in (Arun et al. [1987]). It can be seen in the Fig. 3.4(b) and (c) that our method maintains robustness while the original method deteriorates. It is worth noting that the general ℓ_1-norm regression based method cannot help if only two plane pieces are present in the scene



(a) 2 modes case. (b) 3 modes case. (c) 4 modes case. (d) Anti-outlier performance.

Figure 3.4: Robustness of the rotation estimation. (a) (b) and (c) compare the performance of the least-squares and the ℓ_1-norm regression based methods for the case of 2, 3 and 4 modes, respectively. Note that in (a), the red line and the green line coincide with each other. The horizontal axes of (a), (b), and (c) denote the standard deviation of the noise that is imposed on the "badly tracked mode". (d) demonstrates the outlier resilience of the two methods for an increasing outlier fraction (10 modes in total). All the results (rotation error under each noise level and outlier number) are the average of 1000 trials with combination of arbitrary bundle structure and groundtruth rotation.

(cf. Fig. 3.4(a)). It is not possible to solve for the rotation with less than two robustly perceived planar modes being observed, as this represents the minimal case.

The second part of our simulation experiments is shown in Fig. 3.4(d), where we register bundles of 10 modes. This experiment evaluates the overall outlier-resilience by perturbing an increasing amount of modes by heavy noise. We compare the performance of our ℓ_1-extension against Arun's original solution. As can be observed, the rotation error stays rather low if at least 50% of the modes are tracked with moderate noise only. This phenomenon confirms the common observation that the ℓ_1-norm scheme can resist up to about 50% of outliers.

3.4.3 Evaluation on a Synthetic Dataset

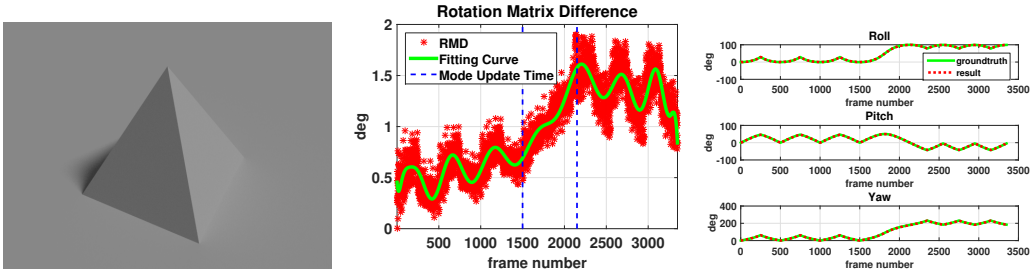
We created a synthetic dataset using the open-source 3D computer graphics software *Blender* to demonstrate two important properties of our algorithm:

1. Piece-wise drift-free performance between bundle or reference updates.

2. Ability to compensate drift when a previously discovered mode is revisited.

The scene in the dataset is composed of a pyramid with four faces on a ground plane. Two types of sensor motion are added to individually confirm the above two properties. In the first case, the sensor orbits in a back-and-forth fashion around the pyramid while the principal axis of the depth camera keeps pointing towards the centre of the pyramid. In the second case, the sensor orbits smoothly and continuously for several complete loops around the pyramid. The groundtruth depth map and the trajectory of the camera are given each time. Realistic noise is added to the depth map before extracting the surface normal vectors.

The dataset and the results concerning the first property are shown in Fig. 3.5. The blue dashed lines in Fig. 3.5 (b) divide the sequence into three parts. They represent the time instants when reference bundle updates happen. We can see that our algorithm returns piece-wise drift



(a) Pyramid. (b) Rotation estimation error. (c) Estimation of roll, pitch and yaw.

Figure 3.5: Performance evaluation on the synthetic dataset "Pyramid". (a) shows the synthetic scene which contains a ground plane and the four faces of a pyramid. The rotation estimation error is shown in (b). The estimated roll, pitch, and yaw angles are shown in (c).

free performance in parts 1 and 3 during which no bundle updates happen, meaning that modes are neither dying nor discovered. The drift keeps increasing in the middle part between the dashed lines, where only one planar mode is robustly tracked. As explained in Section 3.4.2, even the general ℓ_1-norm regression scheme cannot help in this situation because only one planar mode is tracked without gross errors.

The results of the long-term drift experiment are illustrated in Fig. 3.6. The two subfigures show the rotation estimation performance of the proposed algorithm without and with the mode memory scheme, respectively. In the first figure, the stair-behaviour again shows the piece-wise drift-free performance, however, an accumulated drift over a longer term exists. In the second figure, we can clearly see that the long-term drift stays bounded as soon as at least one of the pyramid surfaces has been revisited for the first time (i.e. after the completion of the first loop).

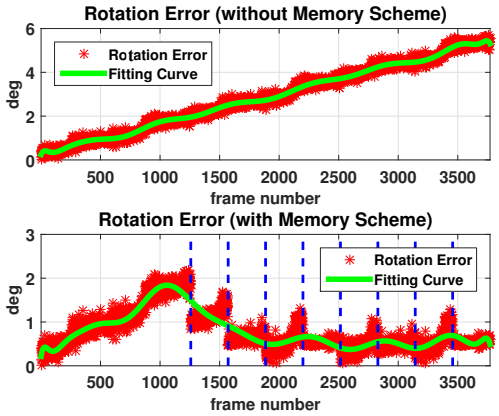


Figure 3.6: The rotation estimation performance of the proposed algorithm without and with the mode memory scheme. An obvious step-like curve in the top figure again demonstrates the piece-wise drift-free behavior. The long-term drift compensation is shown in the bottom figure, where the blue dashed lines denote the time instants when planar modes are revisited and the accumulated rotational drift gets compensated.

3.4.4 Evaluation on Real Data

We compare the performance of our method against two state-of-the-art, open-source motion estimation framework for depth cameras, namely DVO (Kerl et al. [2013b]) and FastICP (Pomerleau et al. [2011]). All methods are evaluated on two published and challenging benchmark datasets from the ETH RGB-D (Pomerleau et al. [2011, 2013]) and TUM RGB-D (Sturm et al. [2012]) series. A qualitative evaluation on the TAMU RGB-D (Lu and Song [2015]) dataset is also given (no groundtruth provided). The datasets we picked for evaluation are listed below and the results are summarized in Table 3.1 as well as illustrated in Fig 3.7.

- ETH 1: 0low_0slow_0fly.
- TUM 1: freiburg3_cabinet.
- TUM 2: freiburg4_structure_texture_near.
- TUM 3: freiburg3_structure_notexture_near.
- TUM 4: freiburg3_structure_notexture_far.
- TAMU 1: corridor_A_const.
- TAMU 2: corridor_B_const.

It is necessary to mention that in some cases our algorithm cannot process the entire sequence.This is due to algorithm limitations that are discussed in the following section. In order to remain fair, we evaluate the performance of all algorithms on the same segments of each sequence. We provide root-mean-square (RMS) and median errors $\tilde{e}$ per second for the rotation estimation. The best performing method's error is each time indicated in bold. It can be seen that our method outperforms both FastICP and DVO in most situations. The relatively bad performance of our method on the ETH 1 dataset is related to the low resolution of this dataset, which leads to a low-quality surface normal vector result. DVO returns a slightly better performance on the TUM 4 sequence, in which plenty of distinctive texture can be observed. Missing numbers in Table 3.1 indicate that the algorithm was not able to successfully process the sequence. Our method handles most of the cases, and remains computationally efficient even on depth images with VGA resolution. Our real-time C++-implementation processes frames at 50 Hz on a laptop with 8 cores. While DVO is real-time capable as well, FastICP quickly drops in computational efficiency as the number of the points increases, and ultimately operates far from real-time on VGA imagery (1 Hz).

3.4.5 Limitations and Failure Cases

Limitations and failure cases of the proposed method are listed as follows:

- The initialisation takes about 1 s. The sensor should not be subjected to substantial motion during this period.
- When only one planar structure is present or can be recognized, the registration of the planar modes based rotation estimator does not work.

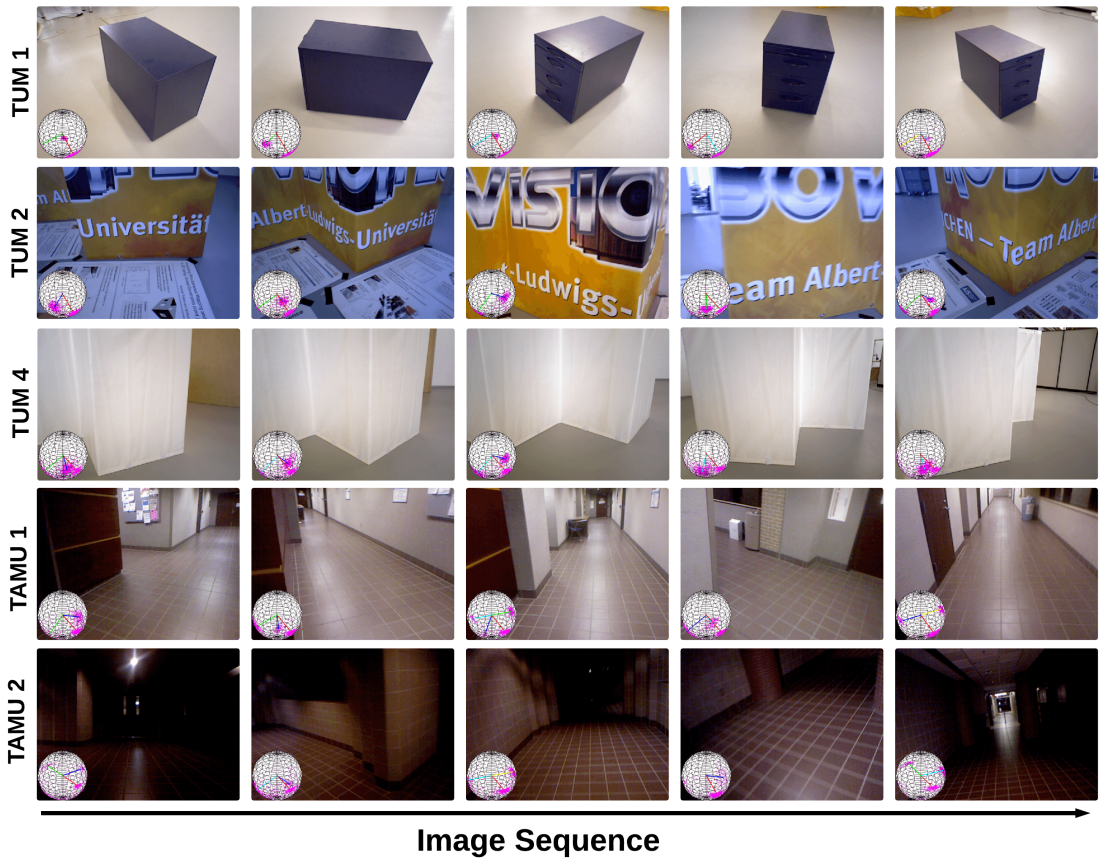


Figure 3.7: Illustration of the proposed algorithm running on a set of real sequences. Note that images shown here are just to illustrate the scenes but are not used in the proposed algorithm. A unit sphere in the bottom-left corner of each image illustrates the planar mode bundle. Corresponding planes in each image of each sequence are denoted with the same color (e.g. the ground plane is always shown in red). We do not show results of TUM 3 because it has a similar scene as TUM 4. We also don't show images for the ETH 1 dataset because it provides only point clouds.

Table 3.1: Performance comparison on several indoor datasets.

Dataset	**DVO**		**FastICP**		**Our Method**	
	$\mathbf{rms}(\mathbf{e}_R)$	$\tilde{\mathbf{e}_R}$	$\mathbf{rms}(\mathbf{e}_R)$	$\tilde{\mathbf{e}_R}$	$\mathbf{rms}(\mathbf{e}_R)$	$\tilde{\mathbf{e}_R}$
ETH 1	×	×	**2.030**	**1.749**	2.892	1.920
TUM 1	4.911	4.456	2.849	1.816	**1.582**	**1.054**
TUM 2	**0.938**	**0.740**	×	×	1.572	1.292
TUM 3	10.898	3.888	8.885	4.920	**1.233**	**0.968**
TUM 4	2.209	1.590	3.674	2.497	**0.983**	**0.683**
Average	4.739	2.669	4.360	2.746	**1.652**	**1.183**

- When two planar modes have a small inscribed angle, the mode seeking may converge to the centre of these two modes and mis-recognize them as a single mode. Such bad initialization can affect the sub-sequent mode tracking iterations as well as the rotation estimation.

3.5 Conclusion

This chapter presented a highly efficient 3D rotation estimation algorithm for depth cameras in piece-wise planar environments. It shows that by using surface normal vectors as an input, planar modes in the corresponding density distribution function can be discovered and continuously tracked using efficient non-parametric estimation techniques. The relative rotation from the reference view to the current view can be estimated by registering entire bundles of planar modes. Robustness of the bundle registration process is achieved by performing a general ℓ_1-norm regression instead of simply solving a least-squares problem. Piece-wise drift-free performance is achieved as long as no bundle updates happen. The chapter furthermore shows that by introducing a mode memory scheme, drift can be avoided even if certain modes are temporally unobserved. Extensive evaluations on simulated, synthetic and real data demonstrate the robustness and effectiveness of the proposed algorithm. Note that our synthetic dataset as well as our code are ready for public release.

The present work unseals an interesting analogy between classical 6 DoF simultaneous localization and mapping (SLAM) of 3D points, and our 3 DoF rotation estimation scheme which shows that—given surface normal vectors—we are able to perform decoupled, simultaneous orientation estimation and mapping of planar modes. In SLAM, long-term drift is eliminated as soon as the 3D points are no longer updated. This corresponds to our drift-free performance in case the reference bundle stays unchanged. Furthermore, our mode-memory scheme has analogies with loop-closure in SLAM, which is well-known to compensate for long-term drift. The analogy with SLAM suggests immediate directions for interesting future work around efficient normal-vector based, decoupled rotation estimation. For instance, we plan to rely on graph-optimization methods leading to a more accurate, multi-frame mode-initialization procedure. Furthermore, the inclusion of appearance information would robustify the reactivation of modes from the memory even in the presence of more significant drift.

Chapter 4

Efficient Density-Based Tracking of 3D Sensors in Manhattan Worlds

Depth sensors produce point cloud measurements. The fundamental problem behind incremental motion estimation with depth sensors therefore is the registration of two point sets A and B. The by far most popular technique is given by the Iterative Closest Point (ICP) method (Besl and McKay [1992]). The basic idea is straightforward: We find approximate correspondences between pairs of points between A and B by simply associating the spatially nearest neighbor of set B to each point of set A. We then minimize the sum of squared distances over a euclidean transformation in closed form. We finally iterate over these two steps until convergence. The complexity of the algorithm is an immediate consequence of the need to find the closest point for each point in each iteration. Even the fastest implementations (Pomerleau et al. [2011, 2013]) therefore fail to deliver real-time performance as soon as we consider modern sensors returning dense depth images at VGA resolution.

The community therefore investigated alternative registration principles that are preceded by a transformation of data into lower dimensional, spatial density distribution functions (Jian and Vemuri [2011]). The general advantage of density alignment based methods is that they do no longer depend on the establishment of one-to-one or even weighted, fuzzy one-to-many point correspondences (Chui and Rangarajan [2000b]). Our work lifts this concept to a general, real-time motion estimation framework for 3D sensors. The key of our approach consists of exploiting the structure of man-made environments, which often contain sets of orthogonal planar pieces. We furthermore rely on efficient dense surface normal vector computation in order to estimate the rotation independently of the translation. As will be shown, the exploitation of this prior furthermore allows us to split up the translational alignment of the density distribution functions into three independent steps, namely one for each dimension.

In summary, a highly efficient motion estimation framework is proposed for popular 3D sensors such as the Microsoft Kinect v.2, based on alignment of density distribution functions. Our contributions are listed as follows:

- Estimation of absolute rotation by exploiting the properties of Manhattan worlds, thus resulting in a manifold-constrained multi-mode tracking scheme.

- Efficient decoupled estimation of individual translational degrees of freedom through 1D kernel density estimates.

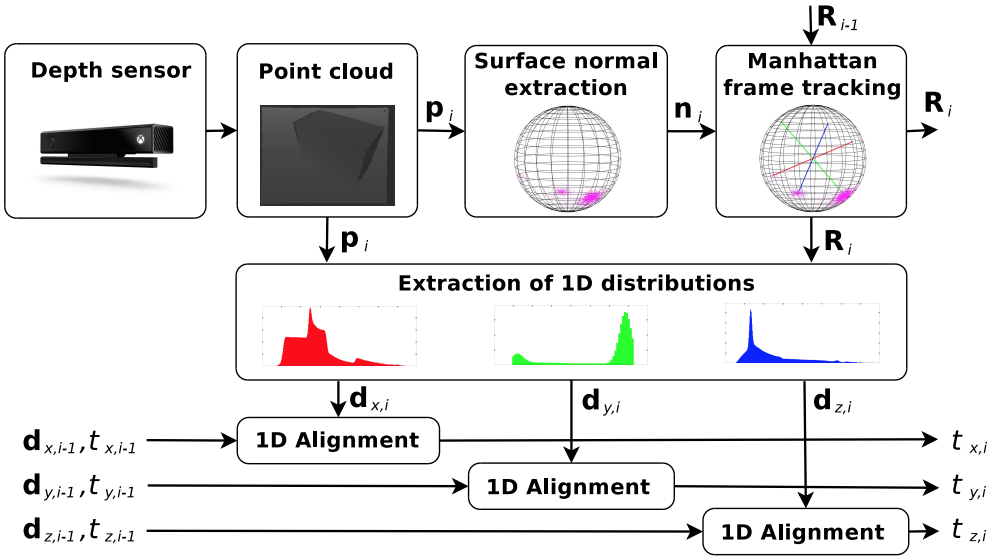


Figure 4.1: Overview of the proposed, decoupled motion estimation framework for 3D sensors in Manhattan worlds.

- Integration into a real-time framework able to process dense depth images with VGA resolution at more than 50Hz on a CPU.

This chapter is organized as follows: We first review related work. Section 4.2 then introduces our main idea for motion estimation in Manhattan worlds based on 3D sensors. The decoupled estimation of rotation and translation are respectively presented in Sections 4.3 and 4.4, respectively. Section 4.5 finally presents our extensive experimental evaluation on both simulated and real data. We test and evaluate our algorithm against existing alternatives on publicly available datasets, showcasing outstanding performance at the lowest computational cost.

4.1 Related Work

Point set registration is a traditional problem that has been investigated extensively in the computer vision community. We are limiting the discussion to methods that process mainly rigid, geometric information. The most commonly used method is given by the ICP algorithm (Besl and McKay [1992]), which performs registration through iterative minimization of the SSD distance between spatial neighbors in two point sets. In order to avoid the costly repetitive derivation of point-to-point correspondences, the community has also investigated the representation and alignment of point clouds using density distribution functions. The idea goes back to Chui and Rangarajan [2000a] and Tsin and Kanade [2004], who represent point clouds as explicit Gaussian Mixture Models (GMM) or implicit Kernel Density Estimates (KDE), and then find the relative transformation (not necessarily Euclidean) by aligning those density distributions. Jian and Vemuri [2011] summarizes the idea of using GMMs for finding the aligning transformation, and notably derives a closed-form expression for computing the L2 distance between two GMMs. Yet another alternative which avoids the establishment of

point-to-point correspondences is given by Fitzgibbon [2003], which utilizes a distance transformation in order to efficiently and robustly compute the cost of an aligning transformation. The distance transformation itself, however, is again computationally intensive.

Classical ICP or even density alignment based methods are prone to local minima as soon as the displacement and thus also the point cloud structure is subjected to too intensive changes. In order to tackle situations of large view-point changes, the community has therefore investigated globally optimal solutions to the point set registration problem, such as (Yang et al. [2013]). These methods are however inefficient and thus not suited for real-time applications, where the frame-to-frame displacement anyway remains small enough for a successful application of local methods.

From a more modern perspective, the ICP algorithm and its close derivatives (Pomerleau et al. [2011, 2013]) still represent the algorithm of choice for real-time LIDAR tracking. The upcoming of RGB-D cameras has however led to a new generation of 2D-3D registration algorithms that exercise a hybrid use of both depth and RGB information. Kerl et al. [2013b] for instance uses the depth information along with the optimized relative transformation to warp the image from one frame to the next, thus permitting direct and dense photometric error minimization. Newcombe et al. [2011b]; Engel et al. [2013, 2014]; Schoeps et al. [2014] apply a similar idea to RGB camera tracking. More recently, Kneip et al. [2015] even apply ICP and distance transforms to semi-dense 2D-3D registration. While the focus of our work is tracking of pure 3D sensors, we evaluate our method on datasets captured by Microsoft Kinect sensors, and thus include a comparison of our results against the method presented in (Kerl et al. [2013b]), which also takes appearance information into account.

The special structure of man-made environments can be exploited to simplify or even robustify the formulation of motion estimation with exteroceptive sensors. Weingarten and Siegwart [2006] and Trevor et al. [2012] introduce planar surfaces into the mapper which are often contained in our man-made environments. Taguchi et al. [2013] combine point and plane features towards fast and accurate 3D registration. In our work, we additionally exploit the fact that indoor environments such as corridors frequently contain orthogonal structure in the surface arrangement. The property was coined as *Manhattan World* (MW) in (Coughlan and Yuille [1999]), where they formulated vanishing point estimation from a single RGB image as a Bayesian inference problem. Košecká and Zhang [2002] present a video compass using a similar idea. Tracking the *Manhattan Frame* can be regarded as absolute orientation estimation, and thus leads to a significant reduction or even complete elimination of the rotational drift. Silberman et al. [2012] improve MW orientation estimation by introducing depth and surface normal information obtained from 3D sensors. More recently, Straub et al. [2014] propose the inference of an explicit probabilistic model to describe the world as a mixture of Manhattan frames. They employ an adaptive Markov-Chain Monte-Carlo sampling algorithm with Metropolis-Hasting split/merge moves to identify von-Mises-Fisher distributions of the surface normal vectors. In (Straub et al. [2015a]), they adapt the idea to a more computationally friendly approach for real-time tracking of a single, dominant MF. Our work is closely related, except that our mean-shift tracking scheme (Fukunaga and Hostetler [1975]) is simpler and more computationally efficient than the MAP inference scheme presented in (Straub et al. [2015a]), which depends on approximations using the Karcher mean in order to achieve real-time performance. We furthermore extend the idea to full 6DoF motion estimation.

4.2 Overview of the Proposed Algorithm

Our method is summarized in Figure 4.1, and consists of three main steps:

- Using the same method as performed in Chapter 3, we extract surface normal vectors $\mathbf{n}_i$ from the measured point clouds, which later allows us to compute the orientation of the sensor independently of the translation.
- We then rely on the assumption that there is a dominant MF in the environment. This allows us to simply track a number of modes in the density distribution of the surface normal vectors, which can be done in a non-parametric way by employing the mean shift algorithm on the unit sphere. It prevents us from having to identify the parameters of a complete explicit model of the density distribution function. We present a manifold-constrained mean-shift algorithm that takes the orthogonal prior into account. Note that the optimization of the rotation is not a classical registration step, but a simple tracking procedure that takes only the information of a single frame into account in order to come up with a drift-free estimate of the absolute orientation.
- By knowing the absolute orientation in each frame, we can easily unrotate the point clouds of a frame pair and assume that the transformation that separates them is a pure translation. A further beneficial consequence is that the principal directions of a Gaussian Mixture Model of the point cloud can be constrained to align with the basis axes. In other words, the covariance matrices become diagonal by which the purely translational alignment cost can effectively be split up into three independent terms, namely one for each dimension. We are therefore allowed to simply solve for each translational degree of freedom independently. We notably do so by extracting kernel density distributions of the point clouds projected onto the basis axes, and by performing three simple 1D alignments. Again note that—due to the unrotation—the obtained relative displacement is immediately expressed in the world frame.

We will in the following explain the details of the rotation and translation alignment.

4.3 Absolute Orientation Based on Manifold-Constrained Mean-Shift Tracking

We estimate the absolute orientation by tracking a dominant Manhattan Frame (MF) in the surface normal vector distribution of each frame. We will start by introducing the mean-shift tracking scheme that operates under the assumption that a sufficiently close initialization point is known. We then conclude by explaining the initialization in the very first frame, which builds on top of our mean-shift extension.

4.3.1 Basic Idea

For structures that obey the Manhattan World (MW) assumption, the surface normal vectors $\mathbf{n}_i$ have an organized distribution on the unit sphere $\mathbb{S}^2$, which can be exploited for recognizing

the MW orientation. It is reasonable to assume that the unit vectors $\mathbf{n}_i$ are samples of a probability density function, as they are more likely to be distributed around the basis axes of the MW (in both directions). The process of finding the dominant axes is therefore equivalent to mode seeking in this density distribution (i.e. finding local maxima in the density distribution function). The modes are additionally constrained to be orthogonal with respect to each other. We therefore express the MF by a proper 3D rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ of which each column $\mathbf{r}_j$ captures the direction of one of the dominant axes of the MF. Note that this contrasts with (Straub et al. [2014]) where the MF is denoted by six signed axes. Our notation is more compact and natural to deal with, as it immediately expresses the pose of the camera as well. Special care however needs to be taken in order to deal with the non-uniqueness of the representation, as each $\mathbf{r}_j$ could in principle be replaced by its negative (although we ensure that $\mathbf{R}$ always remains a right-handed matrix).

The mean shift based method is applied as in Chapter 3 to seek planar modes. For each basis vector $\mathbf{r}_j$, one mean shift vector is simply computed, which potentially results in a non-orthogonal updated MF $\hat{\mathbf{R}}$. We therefore finish each overall iteration by reprojecting $\hat{\mathbf{R}}$ onto the nearest $\mathbf{R} \in \mathrm{SO}(3)$. To clearly demonstrate the application of the mean-shift method in the case of MW, the following first revisits the update of each mode within a single mean-shift iteration, then discusses the projection back onto SO(3).

4.3.2 Seeking the Dominant Axes

The mean-shift iteration for a dominant axis given a set of normal vectors on $\mathbb{S}^2$ works as follows:

- First we find all normal vectors that are within a neighbourhood of the considered centre $\mathbf{r}_j$. The window is a conic section of the unit sphere and the apex angle of the cone θ_{window} defines the size of the local window. Relevant normal vectors for mode j need to lie inside the respective cone, and thus pass the condition

$$\|\mathbf{n}_i \times \mathbf{r}_j\| < \sin(\frac{\theta_{\text{window}}}{2}). \tag{4.1}$$

  Let us define the index i_j which iterates through all $\mathbf{n}_i$ that fulfill the above condition. Note that—if choosing $\theta_{\text{window}} < \frac{\pi}{2}$—every $\mathbf{n}_i$ contributes to at most one mode.

- Then all contributing $\mathbf{n}_{i_j}$ are projected into the tangential plane at $\mathbf{r}_j$ in order to compute a mean shift. Let

$$\mathbf{Q} = \begin{bmatrix} \mathbf{r}_{mod(j+1,3)} & \mathbf{r}_{mod(j+2,3)} & \mathbf{r}_{mod(j+3,3)} \end{bmatrix}. \tag{4.2}$$

  Then

$$\mathbf{n'}_{i_j} = \mathbf{Q}^T \mathbf{n}_{i_j} \tag{4.3}$$

  represents the normal vector rotated into the MF, with a cyclic permutation of the coordinates such that the last coordinate is along the direction of axis j. A transformation similar to the Riemann exponential map is applied in order for the distances in the tangential plane to represent proper geodesics on $\mathbb{S}^2$ (or equivalently angular deviations).

The rescaled coordinates in the tangential plane are given by

$$\begin{aligned} \mathbf{m'}_{i_j} &= \frac{\sin^{-1}(\lambda)\,\mathrm{sign}(n'_{i_j,z})}{\lambda} \begin{bmatrix} n'_{i_j,x} \\ n'_{i_j,y} \end{bmatrix}, \\ \text{where } \lambda &= \sqrt{n'^2_{i_j,x} + n'^2_{i_j,y}}. \end{aligned} \tag{4.4}$$

Note that this projection has the advantage of correctly projecting normal vectors from either direction into the same tangential plane.

- We compute the mean shift in the tangential plane

$$\mathbf{s'}_j = \frac{\sum_{i_j} e^{-c\|\mathbf{m'}_{i_j}\|^2} \mathbf{m'}_{i_j}}{\sum_{i_j} e^{-c\|\mathbf{m'}_{i_j}\|^2}}. \tag{4.5}$$

c is a design parameter that defines the width of the kernel.

- Finally, the mean shift is transformed back onto the unit sphere using the Riemann logarithmic map, and then it is rotated back into the camera frame

$$\mathbf{s}_j = \mathbf{Q} \overline{\begin{bmatrix} \frac{\tan(\|\mathbf{s'}_j\|)}{\|\mathbf{s'}_j\|} \mathbf{s'}_j^T & 1 \end{bmatrix}}^T, \tag{4.6}$$

where $\overline{[\cdot]}$ returns the input 3-vector divided by its norm.

4.3.3 Maintaining Orthogonality

After computing a mean shift for each mode $\mathbf{r}_j$, we effectively obtain an expression for the updated rotation matrix

$$\hat{\mathbf{R}} = \begin{bmatrix} \hat{\mathbf{r}}_0 & \hat{\mathbf{r}}_1 & \hat{\mathbf{r}}_2 \end{bmatrix}, \text{where} \tag{4.7}$$

$$\hat{\mathbf{r}}_j = \mathbf{r}_j + \mathbf{s}_j, j = 0, 1, 2. \tag{4.8}$$

This update may however violate the orthogonality constraint on our rotation matrix. We easily circumvent this problem by re-projecting $\hat{\mathbf{R}}$ onto the closest matrix on $SO(3)$ under the Frobenius norm. If

$$[\mathbf{U}, \mathbf{D}, \mathbf{V}] = \mathrm{SVD}(\hat{\mathbf{R}}), \tag{4.9}$$

the final updated rotation matrix is easily given by

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T. \tag{4.10}$$

As illustrated in Figure 4.2, our method thus represents a double, cascaded manifold-constrained mean-shift extension, where the update of each mode is enforced to remain on the $\mathbb{S}^2$ manifold, and the combination of all three modes is each time enforced to remain an element on the $SO(3)$ manifold. In other words, in each iteration we compute the $SO(3)$-consistent update that is closest to the individual mean-shift updates.

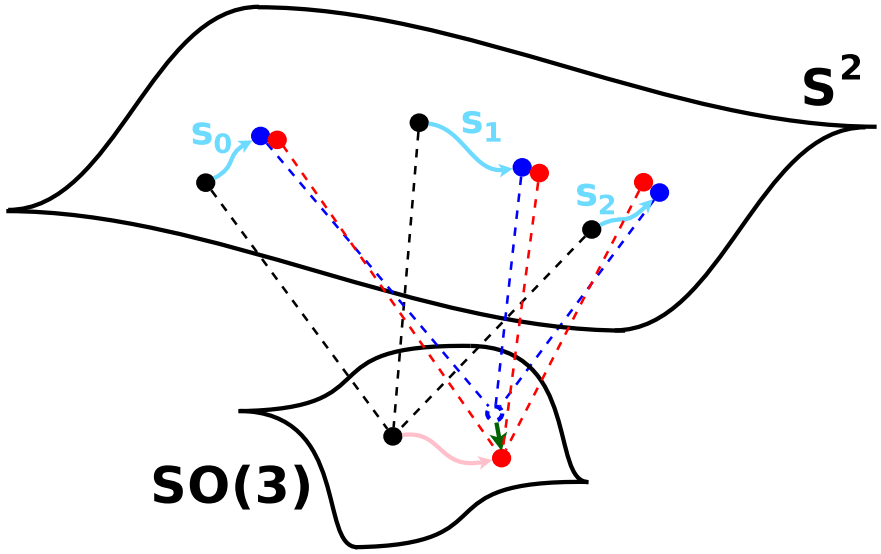


Figure 4.2: Illustration of our cascaded manifold-constrained mean-shift implementation. We first compute updates $\mathbf{s}_j$ for each mode on $\mathbb{S}^2$, which brings us from the black to the blue modes. The blue modes however do no longer represent a point on the underlying manifold $SO(3)$. We find the nearest rotation through a projection onto the manifold (green arrow), thus returning the red modes which are closest and at the same time fulfill the orthogonality constraint.

4.3.4 Initialization in the First Frame

We use mean-shift clustering to initialize the algorithm, and thus build on top of our MF tracking scheme. The procedure is summarized in Figure 4.3. We simply run the MF tracking procedure for 100 times, each time starting from a random initial rotation. This returns a redundant set of candidate MFs, within which we need to identify the most dominant cluster in order to complete the initialization. In fact, typically only a very small number of trials will not converge to the dominant MF if there is only one MF in the observed scene. However, the MF estimates are not directly comparable since one and the same MF may indeed be found or represented by any permutation or negation of individual basis vectors, as long as the result remains a right-handed matrix. In fact, there are 24 possible representations for one and the same MF. In order to render the results comparable and identify the dominant MF cluster, we convert the matrices into a canonical form based on a set of simple rules. For instance, the number of possible representations can already be reduced to 4 by simply requiring the basis vector with the potentially highest z-coordinate to be the one corresponding to the z-axis. To finally identify the dominant cluster, we simply group them based on a simple distance metric between rotation matrices, as well as a fixed threshold.

4.4 Translation Estimation through Separated 1-D Alignments

Taking advantage of the Manhattan World properties, the translation in each dominant direction can be estimated separately. In this section, the 1D alignments that rely on kernel density

distribution functions are first discussed. A convergence analysis is given in the following.

4.4.1 Independence of the Three Translational Degrees of Freedom

Although we are not using an explicit model for representing the density distributions, let us assume for a moment that it is given by a simple Gaussian (i.e. a toy GMM) to see the implications of a Manhattan world and a known absolute orientation of the Manhattan frame. A Gaussian in 3D with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is simply given by:

$$\phi(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{\exp[-0.5(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})]}{\sqrt{(2\pi)^3|\det(\boldsymbol{\Sigma})|}}. \tag{4.11}$$

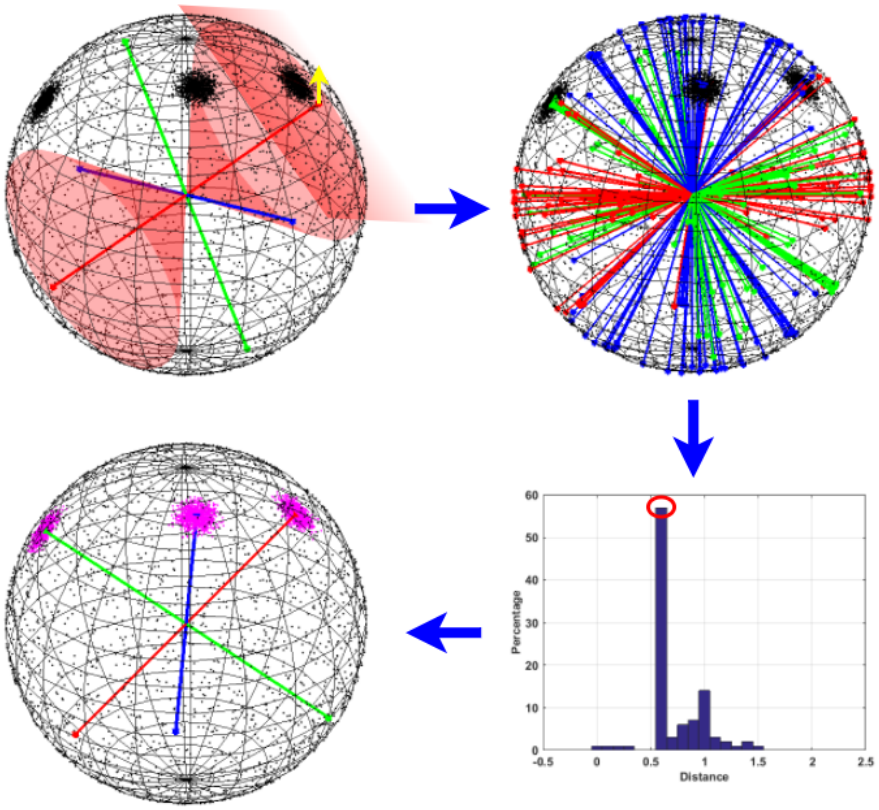


Figure 4.3: The mechanism of the initial Manhattan frame seeking. The first figure shows the start from a random rotation. Each dominant direction is refined by performing a mean-shift iteration on the tangential space. The second figure shows the redundant result obtained by tracking 100 times from random starts. The redundancy of the estimated rotation matrices $\mathbf{R}$ is removed by first converting all the $\mathbf{R}$ to canonical form followed by a histogram-based non-maximum suppression. The final result is shown in the fourth figure. For the sake of clear visualization, the illustrated example contains a significant part of uniformly distributed noisy normal vectors. Note that the proposed seeking strategy is even able to find multiple MFs in the environment, and thus come up with a mixture of Manhattan frames.

There are two Gaussians in two frames and—using the known absolute orientations to unrotate the point clouds—they are separated by a pure translation $\mathbf{t}$. By adding $\mathbf{t}$ to the mean of the Gaussian in the second frame, the kernel correlation between the two Gaussians can be calculated by:

$$\begin{aligned} \mathrm{D} &= \int \phi(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\phi(\mathbf{x}|(\boldsymbol{\mu}_2 + \mathbf{t}), \boldsymbol{\Sigma}_2)d\mathbf{x} \\ &= \phi(\mathbf{0}|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 - \mathbf{t}, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2). \end{aligned} \tag{4.12}$$

We now simplify the case by assuming that the unrotated point clouds can be expressed by a 3D Gaussian distribution with a diagonal covariance matrix. This is reasonable since the unrotated point clouds will indeed contain sets of points that are parallel to the basis axes. Let $\boldsymbol{\Sigma}_d = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 = \mathrm{diag}(\sigma_{dx}, \sigma_{dy}, \sigma_{dz})$, and $\boldsymbol{\mu}_d = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. Then the kernel correlation becomes

$$\begin{aligned} \mathrm{D} &= \frac{\exp[-0.5(\frac{(t_x-\mu_{dx})^2}{\sigma_{dx}} + \frac{(t_y-\mu_{dy})^2}{\sigma_{dy}} + \frac{(t_z-\mu_{dz})^2}{\sigma_{dz}})]}{\sqrt{(2\pi)^3\sigma_{dx}\sigma_{dy}\sigma_{dz}}} \\ &= k \cdot e^{\frac{(tx-\mu_{dx})^2}{-2\sigma_{dx}}} e^{\frac{(t_y-\mu_{dy})^2}{-2\sigma_{dy}}} e^{\frac{(t_z-\mu_{dz})^2}{-2\sigma_{dz}}}. \end{aligned} \tag{4.13}$$

The goal of the alignment in this toy example is to find $\mathbf{t}$ such that D is maximized. It is clear that the above expression involves the product of three independent and positive elements, which means that maximizing each one independently will also maximize the overall distance between the Gaussians.

4.4.2 Alignment of Kernel Density Distribution

Now we move over to our translation alignment procedure, which relies on implicit kernel density distribution functions. Assuming that the absolute orientation with respect to the MF is given, thus each degree of freedom can be solved independently, as in our toy GMM-based example. We therefore compensate for the absolute rotation of the point clouds, and project them onto each basis axis to obtain three independent 1D point sets. Inspired by popular point-set registration works, we then express the 1D point sets via kernel density distribution functions. We sample the function at regular intervals between the minimal and the maximal value. A Gaussian kernel with constant width is used to extract the density at each sampling position. Finally, the alignment between pairs of discretely sampled 1D signals seeks the 1D shift that minimizes the correlation distance between the two signals. It is worth to note that minimizing the correlation distance is equivalent to maximizing the kernel correlation as discussed above. The correlation distance for each pair of 1-D discrete signals is defined as

$$\mathcal{F} = \sum_{i=1}^{n} \left(f(x_i + t) - g(x_i)\right)^2, x_i \in X, \tag{4.14}$$

where X denotes a set of sampling positions for which a density is extracted using a Gaussian kernel. The functions f and g record the density at discrete sampling positions. The correlation distance is the sum over the squared differences at each sampling position. However, the

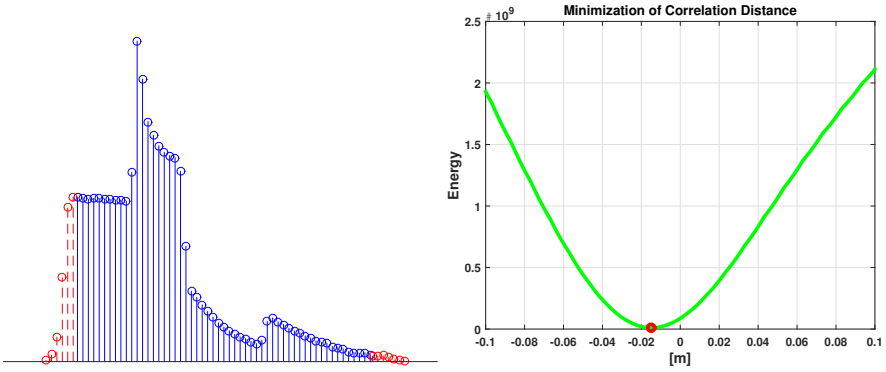


Figure 4.4: The left figure shows an example of discretly sampled distribution truncated on the left and right sides (see red dashed lines). The right figure shows the convergence performance. After the truncation, the minimization problem has only one local minimum with a reasonably large convergence basin.

variable t is continuous which leads to the problem of obtaining density values in between the sampled positions. We solve this problem by employing linear interpolation to obtain values in between neighboring samples.

4.4.3 Convergence Analysis

In order to guarantee the convergence of the minimization of the correlation distance between two discretely sampled distributions f and g, several issues need to be taken into account. First, it is of course vital for f and g to provide density information of the same structure which we call the overlap region. The correlation distance will notably reach its minimum when the overlapping regions align with each other. However, due to the motion of the sensor, the observed structures in successive viewpoints are different, especially along the border of the depth map. This leads to differences in the sampling positions and values. Our solution is to truncate the distribution f, as shown in Fig 4.4. This ensures that the sampling positions of the distribution g fully include the ones of the truncated f.

The second issue occurs when the sensor moves orthogonally to the structure, in which case the sampling density changes. A simple solution is to apply a normalization of the distribution. As we observed during experiments on real data, this is not really needed, except if the sensor moves very close to the structure.

The last issue concerns the choice of the distance function. It is well known that the L_1-distance performs better than the L_2-distance in the presence of outliers. However, there is no noticeable difference in the accuracy of the translation estimation between both norms. This can be attributed to the kernel density distribution alignment, which is robust by nature.

4.5 Experiment

This section evaluates the proposed algorithm. First, the parameter configuration is discussed. Then two simulation experiments are provided to show 1) the robustness of our manifold-

constraint mean-shift based MF seeking algorithm and 2) the benefit to the estimation of translation by aligning the density distribution along the axes of the MF. We conclude with comparing our algorithm against two other state-of-the-art visual odometry solutions on several publicly available datasets. A discussion of the limitations and failure cases of the proposed method is given at last.

4.5.1 Parameter Configuration

In the MF seeking (i.e. the initialization of the absolute rotation), the total number of random starts is set to $N_{trial} = 100$. The apex angle is set to 90° during the initialization and 20° during later tracking. This reduction of the cone apex angle is justified by the assumption that the orientation of the MF does not change too much under smooth motion. Each iterative mean-shift procedure terminates once the angle of the update rotation within one iteration falls below a threshold angle $\theta_{Converge}$, which we set to 1°. The factor c in (4.5) is set to 20. Mean-shift updates are furthermore required to have a minimum number N_{min} of surface normal vectors within the dual-cone. The value of N_{min} depends on the resolution of the input depth map. For low resolution sensors (e.g. Kinect v.1, 160×120), $N_{min} = 30$. For high resolution sensors (Kinect v.2, 640×480), $N_{min} = 100$.

The parameters for the translation estimation contain two parts. The first part concerns the extraction of the density distributions. The sampling between the minimum and maximum value along each basis axis is done in constant intervals of $\delta_s = 0.01m$. The width of the Gaussian kernel for the KDEs is set to $0.03m$. The second part concerns the actual minimization of the correlation distance between each pair of 1D distributions. We simply employ gradient descent with an initial step size of 0.0001m. The search range is furthermore restricted to $\pm 0.1m$.

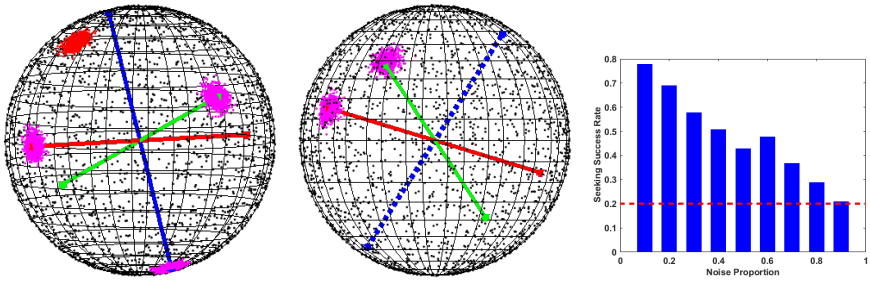
4.5.2 Simulation

4.5.2.1 Manhattan Frame Seeking in Difficult Cases

In this first simulation experiment we show that our manifold-constrained Manhattan frame tracking (including the initialization) can work robustly in challenging cases that may occur on real data as well:

- In the first experiment, the sensor observes additional planar structures for which the normal vector does not align with any of the Manhattan frame's dominant directions. In this case, there will be more than three modes in the distribution on the unit sphere, as shown in Fig 4.5 (a). The three cyan modes represent the MF structure while the red one represents an additional slanted plane. Due to the underlying $SO(3)$ manifold-constrained mean-shift updates, which enforce orthogonality in the mode directions, our algorithm ignores the additional mode and converges to the dominant Manhattan frame.

- Another challenging case is when only two dominant directions of the MF can be observed. In this case, the lost direction can be recovered by exploiting orthogonality and right-handedness between all dominant directions. Fig 4.5 (b) shows an example of such



(a) (b) (c)

Figure 4.5: Robust MF seeking performance in several challenging cases. (a): Seeking the dominant MF when an additional mode/slanted plane exists. (b): Seeking the dominant MF in the case where only two modes can be observed. (c): The success rate of MF seeking under different levels of noise.

a situation. Only the two cyan modes are found by the algorithm, the third direction (indicated with a blue dotted line) is hallucinated.

- In Fig 4.5 (c), we finally demonstrate how the tracking of a MF from a random initial rotation performs under increasing levels of noise. The horizontal axis indicates the overall proportion of noisy normal vectors. It can be observed that as the noise increases, the success rate of the algorithm gradually drops (averaged over many trials). During our initialization procedure, the initial MF orientation is selected from a peak in a histogram over 100 trials. If this peak counts more than a certain threshold percentage of all initialization trials (0.2 in our experiments), the initial MF is likely to be picked up. Therefore, with 100 trials, our algorithm can successfully initialize the MF even if 90% of the normal vectors represent uniformly distributed noise.

4.5.2.2 Translation Estimation in the Manhattan Frame

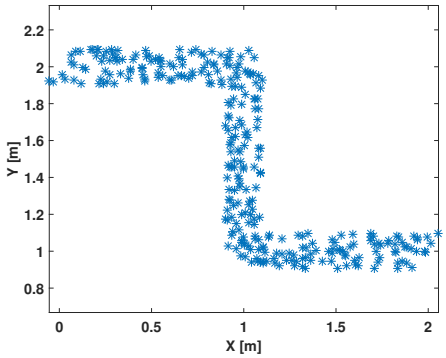
Here we demonstrate the benefit of performing the 1D distribution alignment in the Manhattan frame rather than an arbitrary frame. Without loss of generality, we imagine the two-dimensional example shown in Figure 4.6 (a). It shows the observation of a simple structure which is perturbed by Gaussian noise. The structure aligns with the x or y axis of the Manhattan frame. The observation of two arbitrary sensor viewpoints can be simulated by rotating the original structure inside the plane. Figures 4.6 (c) and (d) show the discrete density distribution along the x-axis of the sensor frame, once from a view-point that is aligned with the Manhattan frame, and once with a rotation of 0.6 rad. It is obvious to see that the distribution inside the Manhattan frame conveys more distinct information than that in an arbitrary sensor view, which is essential for accurate estimation of the translational displacement. The groundtruth displacement in this experiment is 0.1m. Figure 4.6 (b) illustrates the mean alignment error for different sensor frame orientations (each time averaged over various noise levels). It can be observed that error-free estimation can only be performed if the sensor frame is aligned with the Manhattan frame. In other words, the point cloud needs to be unrotated into the Manhattan frame before establishing the 1D density distribution signals and estimating the translation.

Severe differences caused by occlusions would typically represent a challenging case, in which the proposed method may be limited to small baseline scenarios. Nonetheless, it would be seen in the real experiments that the presented approach works effectively.

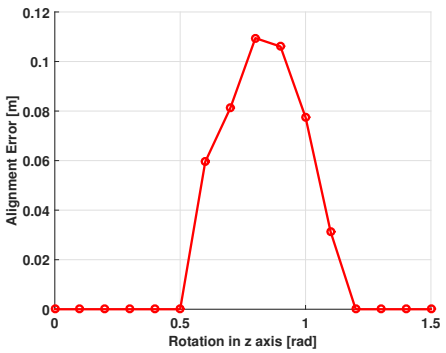
4.5.3 Evaluation on Real Data

We compare the performance of our method to two state-of-the-art, open-source motion estimation implementations for 3D sensors, namely DVO (Kerl et al. [2013b]) and FastICP (Pomerleau et al. [2011]). We evaluate the methods on several challenging benchmark datasets from the TUM RGB-D (Sturm et al. [2012]) and ETH RGB-D (Pomerleau et al. [2011, 2013]) series. The datasets we picked for evaluation are listed below and the results are summarized in Table 4.1.

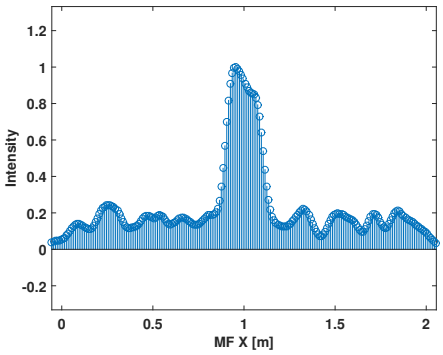
- TUM 1: freiburg3_cabinet.
- TUM 2: freiburg3_structure_notexture_far.
- TUM 3: freiburg3_structure_notexture_near.



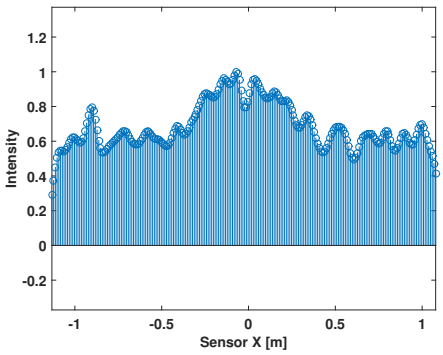
(a) Gaussian noise perturbed structure in MF.



(b) Mean alignment error for different sensor viewing angles.



(c) Density distribution along x-axis of the Manhattan frame.



(d) Distribution along x-axis of a rotated sensor frame.

Figure 4.6: Simulation to demonstrate the benefit of performing the distribution alignment in the MF.

- ETH 1: 0low_0slow_0fly.

It is worth to note that for some of the datasets, our algorithm cannot process the entire sequence due to the limitations that are discussed in the following section. In order to be fair, we evaluate the performance of all algorithms on the same segments of each sequence. A detailed result of the TUM 1 dataset is shown in Figure 4.7. We also provide root-mean-square errors (RMSE) and median errors $\tilde{e}$ per second for both rotation and translation estimation in Table 4.1. The best performing method's error is each time indicated in bold.

It can be seen that in most cases, once the MW assumption is sufficiently met, our result provides very low drift in both rotation and translation. It is outperforming both FastICP and DVO in most situations, and DVO especially in texture-less situations such as the TUM 1 dataset. On the other hand, our method remains computationally efficient even on depth images with VGA resolution, and processes frames at about 50Hz. While DVO is real-time capable as well, FastICP quickly drops in computational efficiency as the number of points increases, and ultimately operates far from real-time on VGA imagery (1 Hz).

4.5.4 Limitations and Failure Cases

The existence of a MW structure in the environment is key to the proposed method. Therefore, the effectiveness of our work currently has the following limitations:

- When only one mode of a MF is observed, the MF tracking will stop and the algorithm terminates.
- When only two orthogonal planes are observed, the MF tracking can continue. However, due to the loss of structural information, the density distribution in the unobserved direction becomes very homogeneous, and the estimation of the respective translation becomes unstable or inaccurate.
- In the case where two MFs on the unit sphere are very close to each other (which could happen in the so-called Atlanta world scenario), our mean-shift tracking may converge in between the two modes, which leads to inaccurate rotation estimation and thus also potentially wrong translation estimation.

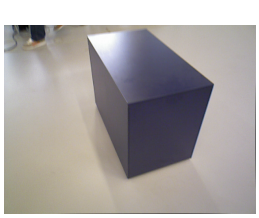
Note that the first two failure cases also affect the ICP algorithm.

4.6 Conclusion

In this chapter, an efficient alternative to the iterative closest points algorithm for real-time tracking of modern depth cameras is presented. We exploit efficient surface normal vector extraction as well as the common orthogonal structure of man-made environments in order to decouple the estimation of the rotation and the three degrees of freedom of the translation. The derived camera orientation is furthermore absolute and thus free of long-term drift. Our method relies on the alignment of density distribution functions, a concept which has linear complexity in the number of points. We therefore achieve not only competitive accuracy, but also superior computational efficiency at the same time.

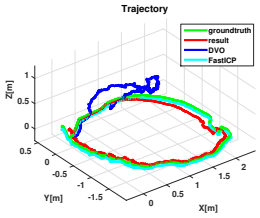
Dataset	**DVO**				**FastICP**				**Our Method**			
	$\mathrm{rms}(e_R)$	$\mathrm{rms}(e_t)$	$\tilde{e}_R$	$\tilde{e}_t$	$\mathrm{rms}(e_R)$	$\mathrm{rms}(e_t)$	$\tilde{e}_R$	$\tilde{e}_t$	$\mathrm{rms}(e_R)$	$\mathrm{rms}(e_t)$	$\tilde{e}_R$	$\tilde{e}_t$
TUM 1	4.911	0.145	4.456	0.128	2.849	0.070	1.816	0.048	**1.308**	**0.030**	**1.037**	**0.022**
TUM 2	2.209	0.102	1.590	0.0620	3.674	0.060	2.497	0.043	**1.063**	**0.031**	**0.759**	**0.020**
TUM 3	10.898	0.197	3.888	0.072	8.885	0.073	4.920	0.048	**1.371**	**0.040**	**1.017**	**0.019**
ETH 1	×	×	×	×	**2.030**	0.509	**1.749**	0.478	2.781	**0.162**	1.875	**0.110**

Table 4.1: Performance comparison on several indoor dataset.

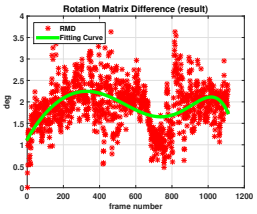


(a) RGB image.

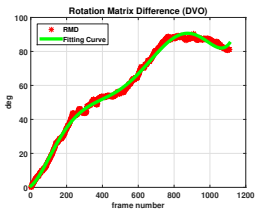
(b) Depth map.



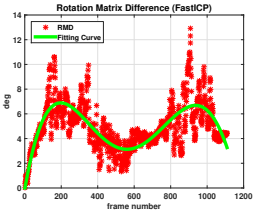
(c) 3D Trajectory.



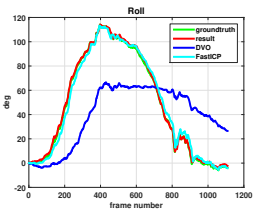
(d) Rotation error of our method.



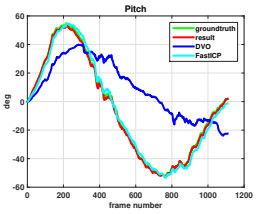
(e) Rotation error of DVO.



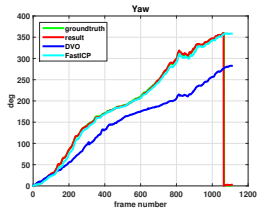
(f) Rotation error of FastICP.



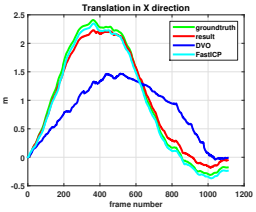
(g) Estimation of roll.



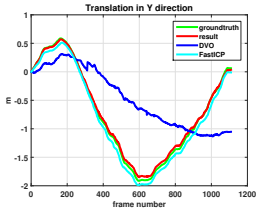
(h) Estimation of pitch.



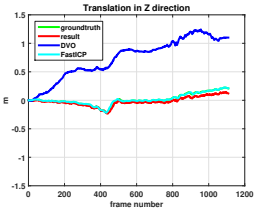
(i) Estimation of yaw.



(j) Translation estimation in x axis of world coordinate.



(k) Translation estimation in y axis of world coordinate.



(l) Translation estimation in z axis of world coordinate.

Figure 4.7: Evaluation of our method on the TUM dataset *cabinet* and comparison to two alternative odometry solutions (FastICP and DVO). Our method (red curve) works at 50 Hz on a CPU for VGA resolution depth images. It outperforms both DVO(blue curve, 30 Hz) and FastICP(cyan curve, 1 Hz) in terms of drift in rotation and translation. More detailed results can be found in Table 4.1

Chapter 5

Visual Odometry with RGB-D Cameras based on Geometric 3D-2D Edge Alignment

In the previous chapter we have seen how to construct an efficient pipeline for RGB-D cameras exploring Manhattan worlds. While this works impressively well, there is also a clear limitation to this approach in terms of the explorable environments. In this chapter, we look at an efficient method to track RGB-D cameras in general environments.

Over the past decade, we have witnessed a number of successful works, such as salient feature based sparse methods (Klein and Murray [2007]; Mur-Artal et al. [2015]), direct methods (Tykkälä et al. [2011]; Steinbrücker et al. [2011]; Audras et al. [2011]; Kerl et al. [2013b]) that employ all intensity information in the image, semi-dense pipelines (Engel et al. [2013, 2014]) and other systems like (Newcombe et al. [2011a]; Whelan et al. [2012b]; Pomerleau et al. [2011, 2013]) which track the camera using an ICP algorithm over the depth information. The present work focusses on edge-based registration, which finds a good compromise between the amount of data used for registration and computational complexity.

Considering that edge detectors have been discovered before invariant keypoint extractors, it comes as no surprise that pioneering works in computer vision such as Larry Robert's idea of a *block's world* (Roberts [1965]) envisage the mapping and registration of entire 3D curves rather than "just" sparse 3D points. While sparse point-based methods have proven to be very effective at subtracting the correspondence problem from the inverse problem of structure from motion, curve-based estimation remains interesting due to the following, geometrically motivated advantages:

- Edges in images make up for a significantly larger amount of data points to be registered to a model, hence leading to superior signal-to-noise ratio and improved overall accuracy.
- Edges represent a more natural choice in man-made environments, where objects are often made up of homogeneously coloured (i.e. texture-less) piece-wise planar surfaces.
- Lines and curves lead to more meaningful 3D representations of the environment than points. Curve-based 3D models may for instance ease the inference of object shapes, sizes and boundaries.

It is the correspondence problem and the resulting computational complexity which however prevented practical, edge or curve-based tracking and mapping pipelines from appearing in the literature until only very recently. Knowing which point from a 3D curve reprojects to which point from a 2D curve measured in the image plane is only easy once the registration problem is solved. Therefore, the correspondence problem has to be solved as part of the 3D-2D registration. Research around the iterative closest point paradigm (Chen and Medioni [1992]), distance transformations (Kiryati and Bruckstein [1996]), and more recent advances such as continuous spline-based parametrisations (Xiao and Li [2005]; Nurutdinova and Fitzgibbon [2015]) nowadays alleviate the iterative computation of putative correspondences, thus rendering online free-form curve-based registration possible.

The contributions of this chapter read as follows:

- A detailed review of 3D-2D free-form edge alignment, summarizing the difficulties of the problem and the solutions given by existing real-time edge alignment methods in robotics.
- Two alternatives to distance transformations — *Approximate Nearest Neighbour Fields* and *Oriented Nearest Neighbour Fields* — with properties that improve the registration in terms of efficiency and accuracy.
- A real-time RGB-D visual odometry system based on nearest neighbour fields, which achieves robust tracking by formulating the 3D-2D ICP based motion estimation as a maximum a posteriori problem.
- An extensive evaluation on publicly available RGB-D datasets and a performance comparison that demonstrates the improvements over previous state-of-the-art edge alignment methods.

The chapter is organized as follows. More related work is discussed in Section 5.1. Section 5.2 provides a review of geometric 3D-2D edge alignment, the problems resulting from employing Euclidean distance fields, and the corresponding solutions of existing methods. Sections 5.3 and 5.4 detail the proposed novel distance transformation alternatives — *Approximate Nearest Neighbour Fields* and *Oriented Nearest Neighbour Fields*. Section 5.5 outlines our complete Canny-VO system with an emphasis on robust weighting for accurate motion estimation in the presence of noise and outliers. Section 5.6 concludes with extensive experimental evaluation.

5.1 Related Work

Curve-based structure from motion has a long-standing tradition in geometric computer vision. Early work by Porrill and Pollard [1991] has discovered how curve and surface tangents can be included into fundamental epipolar geometry for stereo calibration, an idea lateron followed up by Feldmar et al. [1995] and Kaminski and Shashua [2004]. However, the investigated algebraic constraints for solving multiple view geometry problems are known to be very easily affected by noise. In order to improve the quality of curve-based structure from motion, further works by Faugeras and Mourrain [1995] and Kahl and Heyden [1998] therefore looked at special types of curves such as straight lines and cones, respectively.

In contrast to those early contributions in algebraic geometry, a different line of research is formed by works that investigate curve-based structure from motion from the point of view of 3D model parametrisation and optimisation. Kahl and August [2003] are among the first to show complete, free-form 3D curve reconstruction from registered 2D images. Later works then focus on improving the parametrisation of the 3D curves, presenting sub-division curves (Kaess et al. [2004]), non-rational B-splines (Xiao and Li [2005]), and implicit representations via 3D probability distributions (Teney and Piater [2012]). These works, however, mostly focus on the reconstruction problem, and do not use the curve measurements in order to refine the camera poses.

Complete structure-from-motion optimisation including general curve models and camera poses has first been shown by Berthilsson et al. [2001]. The approach however suffers from a bias that occurs when the model is only partially observed. Nurutdinova and Fitzgibbon [2015] illustrate this problem in detail, and present an inverse data-to-model registration concept that transparently handles missing data. Fabbri and Kimia [2010] solve the problem by modelling curves as a set of shorter line segments, and Cashman and Fitzgibbon [2013] model the occlusions explicitly. The successful inclusion of shorter line segments (i.e. edglets) has furthermore been demonstrated in real-time visual SLAM (Eade and Drummond [2009]). Further related work from the visual SLAM community is given by Engel et al. [2013, 2014], who estimate semi-dense depth maps in high-gradient regions of the image, and then register subsequent images based on a photometric error criterion. As common with all direct photometric methods, however, the approach is difficult to combine with a global optimization of structure, and easily affected by illumination changes.

The core problem of projective 3D-to-2D free-form curve registration goes back to the difficulty of establishing correspondences in the data. The perhaps most traditional solution to this problem is given by the ICP algorithm (Chen and Medioni [1992]; Besl and McKay [1991]; Pomerleau et al. [2013]). Yang et al. [2016] even developed a globally optimal variant of the ICP algorithm, which is however too slow for most practically relevant use-cases. Pomerleau et al. [2011] and Tykkälä et al. [2011] present real-time camera pose registration algorithms based on the ICP algorithm, where the latter work minimises a hybrid geometry and appearance based cost function. Both works however cast the alignment problem as a 3D-3D registration problem. More recently, Kneip et al. [2015] show how to extend the idea to 3D-2D registration of edge-based depth maps in a reference frame.

The caveat of the ICP algorithm is given by the repetitive requirement to come up with putative correspondences that still can help to improve the registration. Zhang [1994] investigated how this expensive search can be speeded up by pre-structuring the data in a K-D-tree. The biggest leap with respect to classical ICP was however achieved through the introduction of distance fields (Kiryati and Bruckstein [1996]). Newcombe et al. [2011a] and Bylow et al. [2013] for instance rely on distance fields to perform accurate real-time tracking of a depth sensor. Steinbrücker et al. [2013] furthermore push the efficiency by adaptive sampling of the distance field (Frisken and Rockwood [2000]). More recently, distance-field based registration has also been introduced in the context of 3D-to-2D registration. Kneip et al. [2015] and Kuse and Shen [2016] show the successful use of 2D distance fields for projective registration of 3D curves. The proposed work follows up on this line of research, and proposes a yet more efficient alternative to distance fields for 3D-2D, ICP-based curve registration. The present ori-

ented nearest neighbour fields notably do not suffer from the previously identified registration bias in the case of partially observed models.

5.2 Review of Geometric 3D-2D Edge Registration

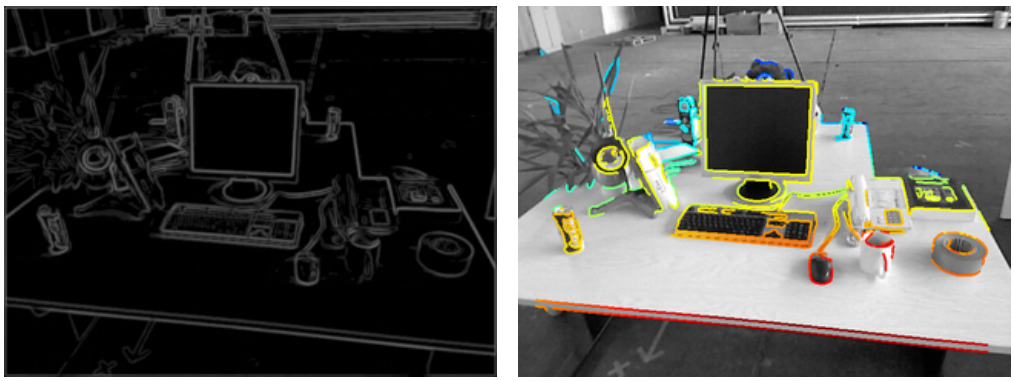
This section reviews the basic idea behind geometric 3D-2D curve alignment. After a clear problem definition, we discuss the limitations of existing Euclidean distance-field based methods addressed through this chapter.

5.2.1 Problem Statement

Let $\mathcal{P}^{\mathcal{F}} = \{\mathbf{p}_i^{\mathcal{F}}\}$ be a set of pixel locations in a frame $\mathcal{F}$ defining the edge map. As illustrated in Fig. 5.1, it is obtained by thresholding the norm of the image gradient, which could, in the simplest case, originate from a convolution with Sobel kernels. Let us further assume that the depth value z_i for each pixel in the edge map is available as well. In the preregistered case, they are simply obtained by looking up the corresponding pixel location in the associated depth image. For each pixel, a local patch (5×5 pixels) is visited and the smallest depth is selected in the case of a depth discontinuity[1]. This operation ensures that we always retrieve the foreground pixel despite possible misalignments caused by extrinsic calibration errors (between the depth camera and the RGB camera) or asynchronous measurements (RGB and depth) under motion. An exemplary result is given in Figure 5.1(b). We furthermore assume that both the RGB and the depth camera are fully calibrated (intrinsically and extrinsically). Thus, we have accurate knowledge about a world-to-camera transformation function $\pi(\lambda\mathbf{f}_i) = \mathbf{p}_i$ projecting any point along the ray defined by a unit vector $\mathbf{f}_i$ onto the image location $\mathbf{p}_i$. The inverse transformation $\pi^{-1}(\mathbf{p}_i) = \mathbf{f}_i$ which transforms points in the image plane into unit direction vectors located on the unit sphere around the center of the camera is also known. If the RGB image and the depth map are already registered, the extrinsic parameters can be omitted. The discussion will be based on this assumption from now on.

Consider the 3D edge map (defined in the reference frame $\mathcal{F}_{\text{ref}}$) as a curve in 3D, and its projection into the current frame $\mathcal{F}_k$ as a curve in 2D. The goal of the alignment step is to retrieve the pose at the current frame $\mathcal{F}_k$ (namely its position $\mathbf{t}$ and orientation $\mathbf{R}$) such that the projected 2D curve aligns well with the edge map $\mathcal{P}^{\mathcal{F}_k}$ extracted in the current frame $\mathcal{F}_k$. Note that—due to perspective transformations—this is of course not a one-to-one correspondence problem. Also note that we parametrize our curves by a set of points originating from pixels in a reference image. While there are alternative parameterizations (e.g. splines), the objective function outlined in this work will remain applicable to any parametrization of the structure.

[1] The depths of all pixels in the patch are sorted and clustered based on a simple Gaussian noise assumption. If there exists a cluster center that is closer to the camera, the depth value of the current pixel will be replaced by the depth of that center. This circumvents resolution loss and elimination of fine depth texture.



(a) Image gradient's norm map. (b) 3D edge map.

Figure 5.1: Image gradients are calculated in both horizontal and vertical direction at each pixel location. The euclidean norm of each gradient vector is calculated and illustrated in (a) (brighter means bigger while darker means smaller). Canny-edges are obtained by thresholding gradient norms followed by non-maximum suppression. By accessing the depth information of the edge pixels, a 3D edge map (b) is created, in which warm colors mean close points while cold colors represent faraway points.

5.2.2 ICP-based Motion Estimation

The problem can be formulated as follows. Let

$$\mathcal{S}^{\mathcal{F}_{\text{ref}}} = \left\{ \mathbf{s}_i^{\mathcal{F}_{\text{ref}}} \right\} = \left\{ d_i^{\mathcal{F}_{\text{ref}}} \pi^{-1}\left(\mathbf{p}_i^{\mathcal{F}_{\text{ref}}}\right) \right\} \tag{5.1}$$

denote the 3D edge map in reference frame $\mathcal{F}_{\text{ref}}$, where $d_i = \frac{z_i}{\mathbf{f}_{i,3}}$ denotes the distance of point $\mathbf{s}_i$ to the optical center. Its projection onto the current frame $\mathcal{F}_k$ results in the edge map

$$\mathcal{O}^{\mathcal{F}_k} = \left\{ \mathbf{o}_i^{\mathcal{F}_k} \right\} = \left\{ \pi\Big(\mathbf{R}^{\mathrm{T}}\big(\mathbf{s}_i^{\mathcal{F}_{\text{ref}}} - \mathbf{t}\big)\Big) \right\}. \tag{5.2}$$

We define

$$n(\mathbf{o}_i^{\mathcal{F}_k}) = \underset{\mathbf{p}_j^{\mathcal{F}_k} \in \mathcal{P}^{\mathcal{F}_k}}{\operatorname{argmin}} \|\mathbf{p}_j^{\mathcal{F}_k} - \mathbf{o}_i^{\mathcal{F}_k}\| \tag{5.3}$$

to be a function that returns the nearest neighbour of $\mathbf{o}_i^{\mathcal{F}_k}$ in $\mathcal{P}^{\mathcal{F}_k}$ under the Euclidean distance metric. The overall objective of the registration is to find

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{o}_i^{\mathcal{F}_k} - n(\mathbf{o}_i^{\mathcal{F}_k})\|^2, \tag{5.4}$$

where $\boldsymbol{\theta} := [c_1, c_2, c_3, t_x, t_y, t_z]^{\mathrm{T}}$ represents the parameter vector that defines the pose of the camera. c_1, c_2, c_3 are Cayley parameters (Cayley [1846]) for orientation $\mathbf{R}$[2], and $\mathbf{t} = [t_x, t_y, t_z]^{\mathrm{T}}$. The above objective is of the same form as the classical ICP problem, which alternates between finding approximate nearest neighbours and registering those putative correspondences, except that in the present case, the correspondences are between 2D and 3D entities. A very similar objective function has been already exploited by Kneip et al. [2015] for robust 3D-2D edge alignment in a hypothesis-and-test scheme. It proceeds by iterative sparse sampling and closed-form registration of approximate nearest neighbours.

5.2.3 Euclidean Distance Fields

As outlined in (Kneip et al. [2015]), the repetitive explicit search of nearest neighbours is too slow even in the case of robust sparse sampling. This is due to the fact that all distances need to be computed in order to rank the hypotheses, and this would again require an exhaustive nearest neighbour search. This is where distance transforms come into play. The explicit location of a nearest neighbour does not necessarily matter when evaluating the optimization objective function (Eq. 5.4), the distance alone may already be sufficient. Therefore, we can pre-process the edge map in the current frame and derive an auxiliary image in which the value at every pixel simply denotes the Euclidean distance to the nearest point in the original edge map. Euclidean distance fields can be computed very efficiently using region growing techniques. Chebychev distance is an alternative when faster performance is required. For further information, the interested reader is referred to (Fabbri et al. [2008]).

Let us define $d(\mathbf{o}_i^{\mathcal{F}_k})$ as the function that retrieves the distance to the nearest neighbour by simply looking up the value at $\mathbf{o}_i^{\mathcal{F}_k}$ inside the chosen distance field. The optimization objective (Eq. 5.4) can now easily be rewritten as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{N} d(\mathbf{o}_i^{\mathcal{F}_k})^2. \tag{5.5}$$

Methods based on Eq. 5.5 cannot provide state-of-the-art performance in terms of efficiency, accuracy and robustness because of the following problems:

- Efficiency: As pointed out by Kuse and Shen [2016], the objective function (Eq. 5.5) is not continuous due to the spatial discretization of the distance field. This problem is bypassed by for example sampling the distances using bi-linear interpolation. However, even with bi-linear interpolation, the distance remains only a piece-wise smooth (i.e. bi-linear) function, as the parametrization changes depending on which interpolation points are chosen. Kuse and Shen [2016] propose to solve this problem by employing the sub-gradient method, which succeeds in the presence of non-differentiable kinks in the energy function. Rather than employing the more popular Gauss-Newton or Levenberg-Marquardt method, they also rely on a less efficient steepest descent paradigm. While

[2]Note that the orientation is always optimized as a change with respect to the previous orientation in the reference frame. The chosen Cayley parametrization therefore is proportional to the local tangential space at the location of the previous quaternion orientation and, therefore a viable parameter space for local optimization of the camera pose.

solving the problem, the bi-linear interpolation and the sub-gradient computation increase the computational burden, and the steepest descent method requires more iterations as the inter-frame disparity becomes larger. To guarantee real-time performance, *e.g.* (Kuse and Shen [2016]) sacrifies accuracy by working on QVGA resolution. In this work, we advocate the use of nearest neighbour fields, which removes the problem of non-differentiable kinks in the energy function.

- Accuracy: As explained in (Nurutdinova and Fitzgibbon [2015]), the model-to-data paradigm is affected by a potential bias in the presence of only partial observations. They propose to replace it by a *data-to-model* concept where the summation runs over the measured points in the image. The work parametrizes curves using B-splines, and an additional curve parameter is required for every data point to define the nearest location on the B-spline. This parameter is simply lifted to an additional optimization variable. Nurutdinova and Fitzgibbon [2015] argue that the *data-to-model* objective is advantageous since it avoids the potentially large biases occurring in the situation of partial occlusions. While the *data-to-model* objective may indeed provide a solution to this problem, it is at the same time a more computational-resource demanding strategy with a vastly blown up parameter space, especially given that the number of automatically extracted pixels along edges can be significantly larger than the number of points in a sparse scenario, and one additional parameter for every data point is needed. Furthermore, the lifted optimization problem in (Nurutdinova and Fitzgibbon [2015]) depends on reasonably good initial pose estimates that in turn permit the determination of sufficiently close initial values for the curve parameters. In this work, we show how an orientation of the field based on the image gradients effectively counteracts this problem while still enabling the more efficient model-to-data paradigm.

- Robustness: Even ignoring the above two problems, a simple minimization of the L2-norm of the residual distances would fail because it is easily affected by outlier associations. In (Kneip et al. [2015]), this problem is circumvented by switching to the L1-norm of the residual distances. In this work, we provide a complete analysis of the statistical properties of the residuals, from which we derive an iterative robust reweighting formulation for 3D-2D curve-registration.

5.3 Approximate Nearest Neighbour Fields

To solve the first problem, we replace the Euclidean distance fields with approximate nearest neighbour fields. As indicated in Figure 5.2, the nearest neighbour field consists of two fields indicating the row and the column index of the nearest neighbour, respectively. In other words, the ANNF simply precomputes the expression $n(\mathbf{o}_i)$ in our optimization objective (Eq. 5.4) for every possible pixel location in the image. Using ANNFs enables us to fix the nearest neighbours during the Jacobian computation, thus removing the problems of discontinuities or non-smoothness during energy minimization. At the same time, the residual evaluation remains interpolation-free, which relieves the computational burden.

From an implementation point of view, it is important to note that the computation of the nearest neighbour field is equally fast as the derivation of the distance field. The reason lies

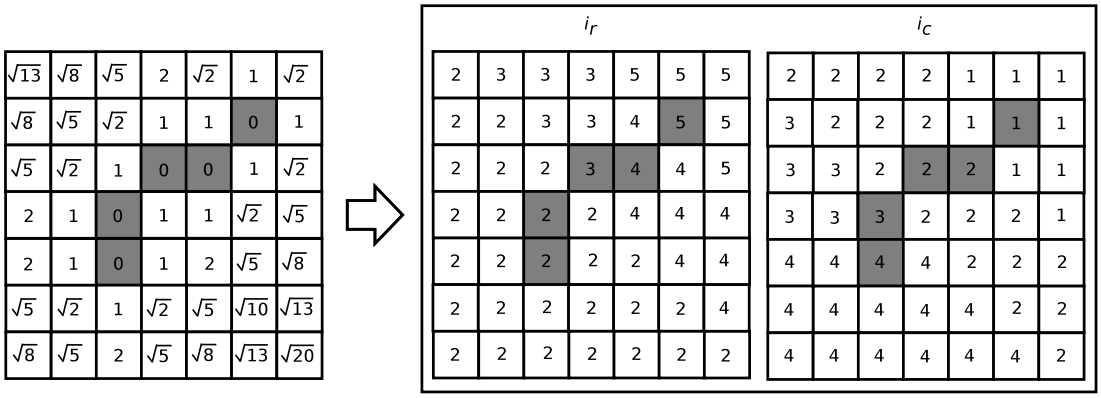


Figure 5.2: Example of a distance field for a short edge in a 7×7 image, plus the resulting nearest neighbour field. i_r and i_c contain the row and column index of the nearest neighbour, respectively.

in the concept of distance field extraction methods (Fabbri et al. [2008]; Felzenszwalb and Huttenlocher [2012]), which typically perform some sort of region growing, all while keeping track of nearest neighbours in the seed region when propagating pixels. Whether we extract a distance field or a nearest neighbour field is merely a question of which information is retained from the computation.

5.3.1 Point-to-Tangent Registration

The ICP algorithm and its variants commonly apply two distance metrics in the registration of 3D point cloud data — the point-to-point distance (Champleboux et al. [1992]) and the point-to-plane distance (Chen and Medioni [1992]). ICP using the point-to-plane distance metric is reported to converge faster than the point-to-point version, especially in the so-called *sliding situation*. In the case of 3D-2D edge alignment, a similar idea to the point-to-plane distance is the point-to-tangent distance. An example is given in Figure 5.3, in which the 2D blue curve is the reprojection of the 3D model while the 2D red curve is the data observed in the current frame. Given a point (green) on the blue curve, the coordinate of its closest point (one of the red points) is returned by the ANNF. The point-to-point residual vector is denoted by $\mathbf{v}_r$ and the point-to-tangent distance is obtained by projecting $\mathbf{v}_r$ to the local gradient direction at the green point. Note that for EDF based methods, only $\|\mathbf{v}_r\|$ is available. Thus, the point-to-tangent distance is not applicable in EDFs. Strictly speaking, the gradient direction needs to be recomputed at the beginning of each iteration. However, as we see through the experiments, the gradient direction of each model point can be assumed constant if there is no big rotation between the reference and the current frame. Note that the image gradient information is already computed during the edge detection process, thus it does not involve any additional computational burden.

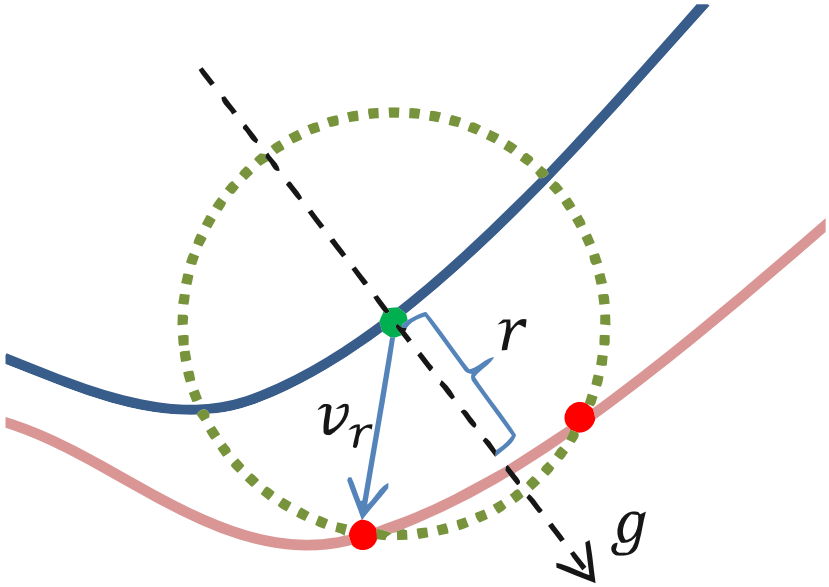


Figure 5.3: Illustration of the point-to-tangent distance. The projected distance r is finally calculated by projecting $\mathbf{v}_r$ onto the direction of the local gradient $\mathbf{g}$.

5.3.2 ANNF based Registration

Using the ANNF, the function $n(\mathbf{o}_i^{\mathcal{F}_k})$ from Eq. 5.3 now boils down to a trivial look-up followed by a projection onto the local gradient direction. This enables us to go back to objective (Eq. 5.4), and we attempt a solution via efficienct Gauss-Newton updates. Let us define the point-to-tangent residuals

$$\mathbf{r} = \begin{bmatrix} \mathbf{g}(\mathbf{p}_1^{\mathcal{F}_{\text{ref}}})^{\text{T}} \left(\mathbf{o}_1^{\mathcal{F}_k} - n(\mathbf{o}_1^{\mathcal{F}_k}) \right) \\ \dots \\ \mathbf{g}(\mathbf{p}_N^{\mathcal{F}_{\text{ref}}})^{\text{T}} \left(\mathbf{o}_N^{\mathcal{F}_k} - n(\mathbf{o}_N^{\mathcal{F}_k}) \right) \end{bmatrix}_{N \times 1}, \tag{5.6}$$

By applying Eq. 5.6 in Eq. 5.4, our optimization objective can be reformulated as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \|\mathbf{r}\|^2. \tag{5.7}$$

Supposing that $\mathbf{r}$ were a linear expression of $\boldsymbol{\theta}$, it is clear that solving Eq. 5.7 would be equivalent to solving $\mathbf{r}(\boldsymbol{\theta}) = \mathbf{0}$. The idea of Gauss-Newton updates (or iterative least squares) consists of iteratively performing a first-order linearization of $\mathbf{r}$ about the current value of $\boldsymbol{\theta}$, and then each time improving the latter by solving the resulting linear least squares problem. The linear problem to solve in each iteration therefore is given by

$$\mathbf{r}(\boldsymbol{\theta}_i) + \left. \frac{\partial \mathbf{r}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i} \boldsymbol{\Delta} = \mathbf{0}, \tag{5.8}$$

and, using $\mathbf{J} = \left.\frac{\partial \mathbf{r}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}$, its solution is given by

$$\boldsymbol{\Delta} = -(\mathbf{J}^{\mathrm{T}}\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{r}(\boldsymbol{\theta}_i). \tag{5.9}$$

The motion vector is finally updated as $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \boldsymbol{\Delta}$.

While evaluating the Jacobian $\mathbf{J}$ in each iteration, the closest points simply remain fixed. This simplification is based on the fact that typically $n(\mathbf{o}_i(\boldsymbol{\theta})) = n(\mathbf{o}_i(\boldsymbol{\theta} + \delta\boldsymbol{\theta}))$ if $\delta\boldsymbol{\theta}$ is a small increment. Furthermore, the equality may not hold when $\mathbf{o}_i$ locates exactly at the border of two pixels. This may lead to gross errors in the Jacobian evaluation, which is why we simply fix the nearest neighbour. The Jacobian $\mathbf{J}$ simply becomes

$$\mathbf{J} = \left[\left(\frac{\partial \left(\mathbf{g}(\mathbf{p}_1^{\mathcal{F}_{\mathrm{ref}}})^{\mathrm{T}} \mathbf{o}_1^{\mathcal{F}_k} \right)}{\partial \boldsymbol{\theta}} \right)^{\mathrm{T}} \quad \cdots \quad \left(\frac{\partial \left(\mathbf{g}(\mathbf{p}_N^{\mathcal{F}_{\mathrm{ref}}})^{\mathrm{T}} \mathbf{o}_N^{\mathcal{F}_k} \right)}{\partial \boldsymbol{\theta}} \right)^{\mathrm{T}} \right]^{\mathrm{T}}_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}. \tag{5.10}$$

Details on the analytical form of the Jacobian are given in Appendix. A.1.1.

5.4 Oriented Nearest Neighbour Fields

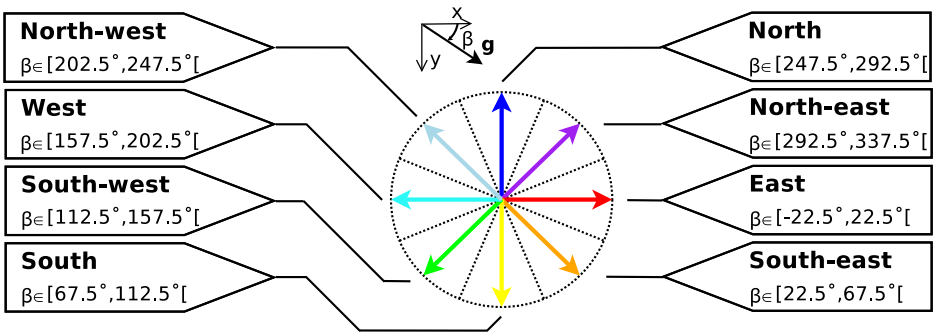
This section explains the idea behind oriented nearest neighbour fields (ONNF) and how they help to improve the performance of model-to-data based projective registration of non-parametric curves. We start by giving a clear definition of the field orientation for distance fields, then show how this design is easily employed to nearest neighbour fields. Finally, a sequence of modifications to this concept is introduced, which gradually improve the accuracy and efficiency of the registration process.

5.4.1 Field Orientation

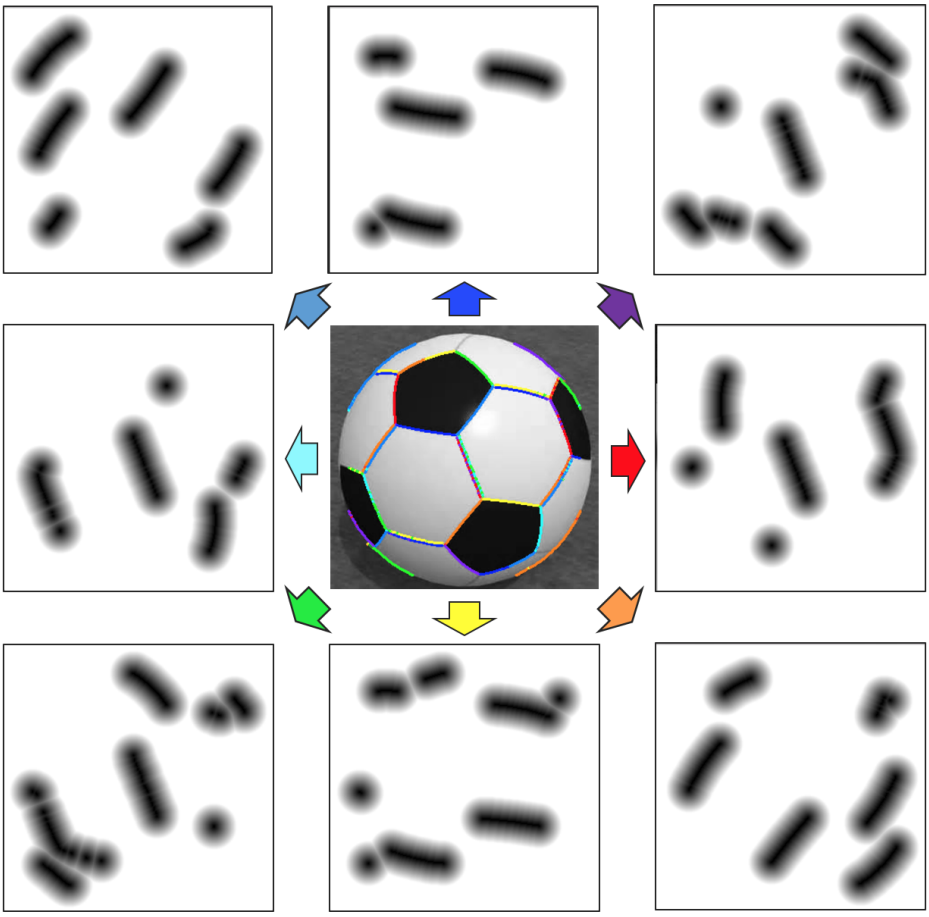
One of the core contributions of this chapter is on orienting the nearest neighbour fields. However, special care is needed to define what *orientation* in the present case means. We explain the concept with distance fields. The most common type of oriented distance field in the 3D computer vision literature is a truncated signed distance field for dense 3D surface reconstruction (Newcombe et al. [2011a]; Bylow et al. [2013]; Steinbrücker et al. [2013]). Given by the fact that the world is always observed from a certain perspective, it makes sense to define the *front* and *back* of a continuous reconstructed surface, which in turn defines the sign of the distances in the field (positive = in front of the surface, negative = behind the surface). In the context of curves in the image, the equivalent would be to define the *inside* and *outside* of contours. This representation, however, would only be unique for a single, closed contour in the image.

A more flexible orientation can be achieved by considering the gradient inclination along the edge. Taking Figure 3 in (Nurutdinova and Fitzgibbon [2015]) as an example, it is also obvious that the registration bias due to partial occlusions in the model-to-data approach could easily be detected or even avoided by considering the "disparity" between the reprojected gradient vector inclinations and the gradient vector inclinations of the nearest neighbours in the

data. We therefore move to oriented distance fields for curves in the image, where the orientation depends on the actual gradient vector inclination.



(a) Discretization bins for gradient vector inclination.



(b) Example oriented distance fields.

Figure 5.4: (a) Orientation bins chosen for the discretisation of the gradient vector inclination (8 bins of 45° width). (b) Example oriented distance fields for edges extracted from an image of a football. Distinct edge segments are associated to only one of the 8 distance fields depending on the local gradient inclination and the corresponding bin.

The idea is straightforward. The distance field is split up into multiple distance fields following a quantisation of the gradient vector inclination. The gradient quantisation adopted is indicated in Figure 5.4(a). It consists of dividing the set of possible gradient vector inclinations into 8 equally wide intervals, each one spanning an angle of 45°. Based on this quantisation table and the local image gradient vector inclination, every pixel along an edge can be associated to exactly one of 8 distance fields. We finally obtain a seed region in each one of 8 distinct distance fields, and can grow each one of them individually, thus resulting in 8 distance fields with exclusively positive numbers (cf. Figure 5.4(b)). Upon registration of a 3D curve, we only need to transform the local gradient of the 3D curve in order to identify the distance field from which the distance to the nearest neighbour of a particular image point has to be retrieved. This formulation has the clear advantage of being less affected by ambiguous associations arising from nearby edges: the distance to the region of attraction of neighbouring edges in the distance field is much larger than in the non-oriented case where all edges appear in the same distance field. In consequence, oriented distance fields also provoque an enlargement of the convergence basin during registration.

Note that the usage of oriented distance fields does not involve any substantial additional computation load. First, the image gradient information is already computed by the edge extraction algorithm. Second, since the complexity of extrapolating a distance field depends primarily on the number of edge points in the seed region, computing the oriented distance fields is similarly fast as computing the non-oriented one. Furthermore, the orientation makes it very easy to parallelise the distance field computation, we merely have to associate one core to each bin of the discretisation.

5.4.2 ONNF based Registration

As shown in Section 5.3, distance fields can be seamlessly replaced by nearest neighbour fields. Thus, the concept of the field orientation is readily applicable to nearest neighbour fields, thus leading to Oriented Nearest Neighbour Fields (ONNF).

Let us define the nearest neighbour in the oriented nearest neighbour field to be

$$\eta_{\mathcal{M}_{\mathcal{G}(\mathbf{o}_i)}}(\mathbf{o}_i) = \underset{\mathbf{m}_j \in \mathcal{M}_{\mathcal{G}(\mathbf{o}_i)}}{\operatorname{argmin}} \|\mathbf{m}_j - \mathbf{o}_i\|_2, \tag{5.11}$$

with $\mathcal{G}(\mathbf{o}_i)$ taking the gradient at the model point corresponding to $\mathbf{o}_i$ and the current camera pose to find the index of the relevant orientation bin (i.e. the index of the relevant nearest neighbour field), and $\mathcal{M}_{\mathcal{G}(\mathbf{r}_i)}$ representing the subset of edge pixels that have fallen into this bin. Similar to what has been proposed in 5.3.2, the residual vectors are projected onto the local gradient direction. Since we are already working in an oriented nearest neighbour filed, this gradient direction is simply approximated by the centre of the corresponding orientation bin, denoted $\mathbf{e}_{\mathcal{G}(\mathbf{o}_i)}$ (as in Figure 5.4(a), the possible $\mathbf{e}_{\mathcal{G}(\mathbf{r}_i)}$ are given by the coloured vectors,

normalised to one). The residuals can finally be defined as

$$\mathbf{r} = \begin{pmatrix} \mathbf{e}^{\mathrm{T}}_{\mathcal{G}(\mathbf{o}_1)} \left(\mathbf{o}_1 - \eta_{\mathcal{M}_{\mathcal{G}(\mathbf{o}_1)}} (\mathbf{o}_i) \right) \\ \vdots \\ \mathbf{e}^{\mathrm{T}}_{\mathcal{G}(\mathbf{o}_n)} \left(\mathbf{o}_n - \eta_{\mathcal{M}_{\mathcal{G}(\mathbf{o}_n)}} (\mathbf{o}_n) \right) \end{pmatrix}, \tag{5.12}$$

and the resulting Jacobian becomes

$$\mathbf{J} = \left[\left(\mathbf{e}^{\mathrm{T}}_{\mathcal{G}(\mathbf{o}_1)} \frac{\partial \mathbf{o}_1}{\partial \boldsymbol{\theta}} \right)^T \quad \dots \quad \left(\mathbf{e}^{\mathrm{T}}_{\mathcal{G}(\mathbf{o}_n)} \frac{\partial \mathbf{o}_n}{\partial \boldsymbol{\theta}} \right)^T \right]^{\mathrm{T}}_{\boldsymbol{\theta}=\boldsymbol{\theta}_k} \tag{5.13}$$

The derivation of the analytical Jacobian is similar to Appendix. A.1.1.

5.4.3 Performance Boost through Adaptive Sampling

Our final modification consists of moving from standard nearest neighbour fields to adaptively sampled nearest neighbour fields (Frisken and Rockwood [2000]). Nearest neighbours at the original image resolution are only computed within a small neighbourhood of the seed region given by the pixels along edges. With reference to Figure 5.5, this corresponds to layer 0. The next step consists of iterating through all edge pixels and keeping track of the closest one to each adjacent location in sub-sampled image grids. Again with reference to Figure 5.5, this corresponds to all higher octaves (i.e. layer 1, layer 2, ...). Note that limiting the filling in higher octaves to adjacent grid locations leads to an implicit truncation of the neighbour field. The concluding step then consists of concatenating the layers by copying the nearest neighbours from all layers to the corresponding locations in the concatenated output matrix, starting from the highest one. Values taken from higher octaves are hence simply overwritten if a lower octave contains more fine-grained information. Figure 5.5 only shows a single nearest neighbour field, but it is clear that the derivation has to be done for each one of the 8 orientation bins, possibly through parallel computation. The adaptively sampled nearest neighbour fields

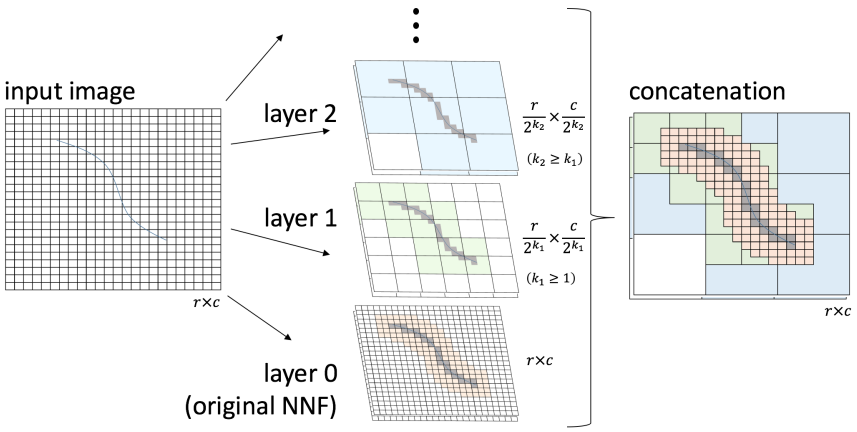


Figure 5.5: Adaptively Sampled Nearest Neighbour Fields. In practice, the concatenated result is just an $n \times m$ matrix where the connected blue and green regions simply contain identical elements.

Table 5.1: Comparison on the properties of different distance transformations

	EDF	ANNF	ONNF
Free of interpolation	×	✓	✓
Enable point-to-tangent distance	×	✓	✓
Enable adaptive sampling	×	✓	✓
Enable registration bias recognition and elimination	×	×	✓

do not involve any loss in accuracy, as the nearest neighbours have maximal resolution within a sufficiently large band around the global minimum. Furthermore, the loss in effective resolution further away from the global minimum does not have a noticeable impact on the ability to bridge even larger disparities. In particular, the fact that the residual vectors are projected onto the direction vector of the corresponding orientation bin causes the approximation error with respect to the exact nearest neighbour to be relatively small. While adaptive sampling is also applicable to distance fields, it would severly complicate the implementation of bi-linear interpolation and hence the definition of continuous residual errors.

A comparison of the properties of all discussed distance transformations is given in Table. 5.1, which helps to highlight the advantages of the proposed distance transformations over the classical Euclidean distance field.

5.5 Robust Motion Estimation

In this section, we discuss how to improve the robustness of the method. A probabilistic formulation is employed in the motion estimation to deal with noise and outliers, which takes the statistical characteristics of the sensor or measurement model into account. Then a simple but effective operation of point culling is introduced, which helps to refine the 3D structure in the reference frame, and thus brings benefits to successive motion estimations. Finally, the whole visual odometry system is outlined.

5.5.1 Learning the Probabilistic Sensor Model

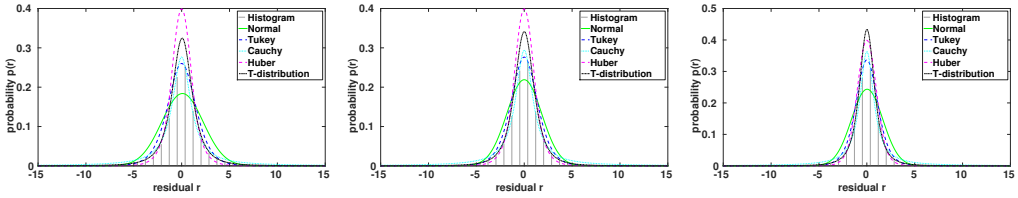
To improve the robustness in the presence of noise and outliers, the motion estimation is formulated as maximizing the posteriori $p(\boldsymbol{\theta}|\mathbf{r})$. Following the derivation in (Kerl et al. [2013b]), the Maximum A Posteriori (MAP) problem is translated into the weighted least squares minimization problem,

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \sum_i \omega(r_i)(r_i(\boldsymbol{\theta}))^2. \tag{5.14}$$

The weight is defined as $\omega(r_i) = -\frac{1}{2r_i}\frac{\partial \log p(r_i|\boldsymbol{\theta})}{\partial r_i}$, which is a function of the probabilistic sensor model $p(r_i|\boldsymbol{\theta})$. IRLS is used for solving Eq. 5.14.

The choice of the weight function depends on the statistics of the residual, which is identified in a dedicated experiment. We investigate several of the most widely used robust weight functions including Tukey[3], Cauchy, Huber (Zhang [1997]) and the T-distribution (Kerl et al.

[3]The Tukey-Lambda distribution is used here rather than the Tukey Biweight function. The closed form of the



(a) Sensor model of *freiburg1*. (b) Sensor model of *freiburg2*. (c) Sensor model of *freiburg3*.

Figure 5.6: Sensor model is obtained by fitting the histogram with different probabilistic distributions.

Table 5.2: Robust weight functions and their parameters fitted on each sub dataset

Type	$\omega(r)$	freiburg1	freiburg2	freiburg3
Tukey	$\begin{cases} \frac{1}{2k^2(e^{r/k}+1)}, \text{if} \lvert r\rvert \leq \epsilon \\ \frac{e^{r/k}-1}{2kr(e^{r/k}+1)}, \text{if} \lvert r\rvert > \epsilon \end{cases}$	k = 0.9609	k = 0.9045	k = 0.7425
Cauchy	$\frac{1}{1+(r/k)^2}$	k = 1.1510	k = 1.0827	k = 0.8754
Huber	$\begin{cases} 1, \text{if} \lvert r\rvert \leq k \\ \frac{k}{\lvert r\rvert}, \text{if} \lvert r\rvert > k \end{cases}$	k = 1.2502	k = 1.2754	k = 1.6999
T-distribution	$\frac{\nu+1}{\nu+(\frac{r}{\sigma})^2}$	ν = 2.2875 σ = 1.1050	ν = 2.7104 σ = 1.0682	ν = 2.4621 σ = 0.8330

[2013b]). The final choice is based on the evaluation in 5.6.2.

5.5.2 Point Culling

Although the probabilistic formulation can deal with noise and outliers, an accurate 3D edge map for each reference frame is still preferred to reduce the risk of an inaccurate registration. Once a new reference frame is created by loading the depth information, the 3D edge map might be not accurate enough because of low-quality depth measurements (e.g. by reflective surfaces) or inaccurate Canny edge detections (e.g. caused by image blur). The successive tracking is possibly affected if the error in the map is not carefully handled. For the sake of computational efficiency, we do not optimize the local map using windowed bundle adjustment as this is commonly done for sparse methods. The number of points used by our method typically lies between 10^3 and 10^4, which is at least one order of magnitude higher than the amount of points used in sparse methods. Therefore, rather than optimizing the inverse depth of such a big number of 3D points, a much more efficient strategy is proposed. All 3D points in the new reference frame are reprojected to the nearest keyframe and those whose geometric residuals are larger than the median of the overall residuals are discarded. We find that this operation significantly improves the accuracy of the motion estimation during the experiments.

Tukey-Lambda distribution requires to set shape parameter $lambda = 0$, which leads to the Logistic distribution. The derivation of the robust weight function is given in Appendix. A.1.2.

5.5.3 Visual Odometry System

Our complete RGB-D visual odometry system is illustrated in Fig. 5.7. There are two threads running in parallel. The tracking thread estimates the pose of the current frame, while the other thread generates new keyframes including the depth initialization. In the tracking thread, only the RGB image of the current frame is used for the Canny edge detection and the subsequent computation of the nearest neighbour field. The objective is constructed and then optimized via the Levenberg-Marquardt method. The reference frame is updated whenever the current frame moves too far away. The distance criterion here is the median disparity between the edges in the reference frame and the corresponding reprojections in the registered current frame. If this value grows larger than a given threshold, a new reference frame is created by the keyframe preparation thread.

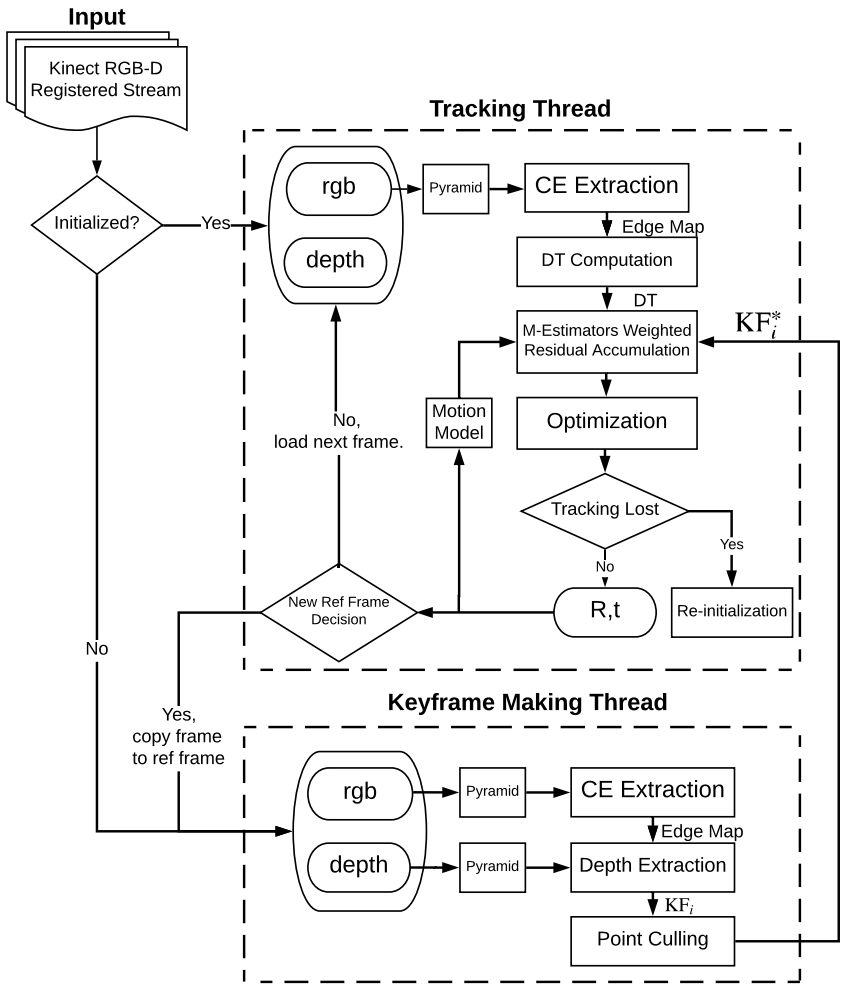


Figure 5.7: Flowchart of the Canny-VO system. Each independent thread is bordered by a dashed line. CE refers to the Canny edge and DT is the abbreviation of distance transformation, which could be one of EDF, ANNF and ONNF.

To deal with large displacement, we apply a pyramidal coarse-to-fine scheme as in (Kerl et al. [2013b]; Engel et al. [2013]) *et al.* . A three-level (from 0 to 2) image pyramid is created. The radius of the distance transformation is adjusted according to the applied level. The registration is performed from the top to the bottom level sequentially. Besides, a motion model is implemented to predict a better starting position for the optimization. This strategy has been widely used in VO and SLAM (Klein and Murray [2007]; Kerl et al. [2013b]; Tanskanen et al. [2013]) and improves the robustness by effectively avoiding local minima in the optimzation. Instead of assuming a prior distribution for the motion as in (Kerl et al. [2013b]), we follow (Klein and Murray [2007]) and implement a simple decaying velocity model, which effectively improves the convergence speed and the tracking robustness.

5.6 Experimental Results

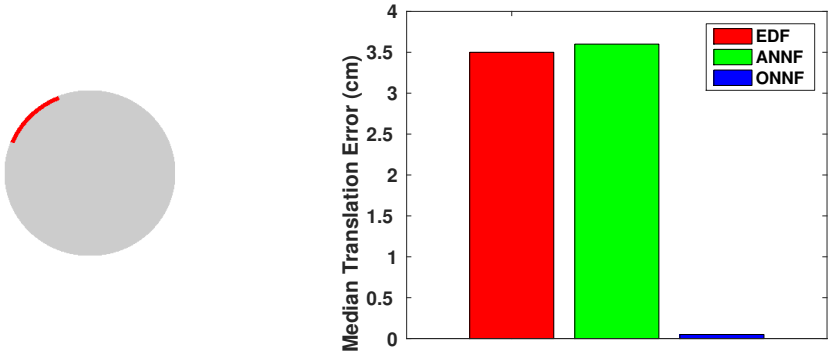
We start with an analysis of the registration bias in the case of partially observed data. We then move over to the optimal parameter choice in the present system, which primarily discusses the choice of the robust weight function. The main experiment compares the quantitative results of trackers that use EDF, ANNF and ONNF, respectively. All variants employ Levenberg-Marquardt optimization. Two publicly available benchmark datasets (Sturm et al. [2012]; Handa et al. [2014]) are used for the evaluation. Finally, we provide a challenging RGB-D sequence to qualitatively evaluate the performance of the present VO system in a relatively large-scale indoor environment.

Note that the trajectory evaluation results listed in the following tables, including relative pose errors (RPEs) and absolute trajectory errors (ATEs) are given as root-mean-square errors (RMSEs). The units for RPEs are $\mathrm{deg}\,/s$ and m/s and the ATEs are expressed in m. The best result is always highlighted in bold.

5.6.1 Handling Registration Bias

The present section contains an important result of this chapter, namely a dedicate experiment on a controlled synthetic sequence to prove the beneficial properties of the presented oriented nearest neighbour fields. We define an artificial circular shape in a reference frame which has a virtual perspective camera model with a focal length of 500.0 and VGA resolution. Let's furthermore assume that the reference view is pointing straight down at the horizontal plane on which the observed image has a width of 28.0 cm. The pose of the reference frame is therefore given by $\mathbf{t} = (0, 0, 218.75)^{\mathrm{T}}$ and $\mathbf{R} = \mathrm{diag}\,(1, -1, -1)$. Once the 3D edge points are extracted, the position of the reference frame is disturbed and re-optimised using either EDF, ANNF or ONNF. To reproduce an example very similar to the one introduced in (Nurutdinova and Fitzgibbon [2015]), only a small continuous part of the circular edge in the image covering $\frac{\pi}{4}$rad is retained (randomly positioned along the circle). Each method is tested for 1000 times. Note that the tests are not using a robust weight function in order not to hide potential biases in the estimation, which is what we are after. Also note that we do not add any noise to the data as the purpose here is to demonstrate the size of convergence basins, numerical accuracy, and estimation biases. As seen in Fig. 5.8, ONNF reports an almost zero bias after optimization,

thus clearly demonstrating its superiority in handling partially observed data over the other two methods.



(a) Circle pattern and the partially observed data. (b) Median translation errors of each method for 1000 trials.

Figure 5.8: Analysis of registration bias in case of only partially observed data.

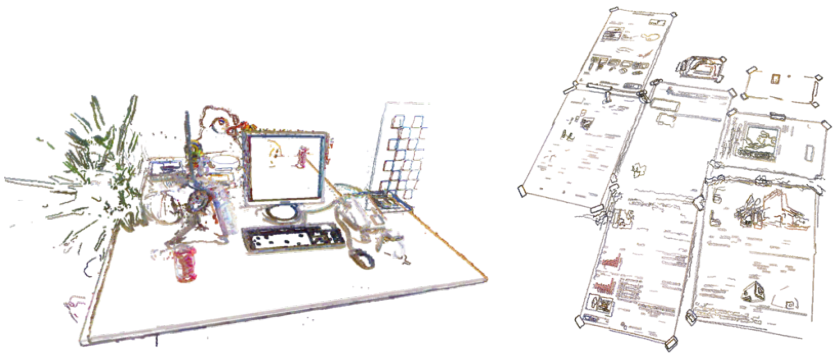
5.6.2 Exploring the Optimal Configuration

An accurate extraction of Canny edges is key to accurate motion estimation. The quality of the gradient map makes the difference. We therefore investigate Sobel filters with different kernel sizes, and find that a symmetric 5×5 kernel outperforms a 3×3 filter and is sufficient for a good estimation. Advance smoothing of the images further helps to improve the edge detection.

To determine the optimal robust weight function, we start by defining reference frames in a sequence by applying the same criterion as in our full pipeline (cf. Fig. 5.7), however using ground truth poses. Residuals are then calculated using the ground truth relative poses between each frame and the nearest reference frame. The residuals are collected over several sequences captured by the same RGB-D camera (*i.e. freiburg 1, freiburg 2, freiburg 3,* respectively), and then summarized in histograms. As an example, all fitting results on the residuals using the ANNF distance metric are illustrated in Fig. 5.6, and the parameters of each model are reported in Table 5.2. The fitting results on the residuals using EDF and ONNF can be obtained in the same way. In general, the T-distribution is the best on fitting the histograms, especially for large residuals.

5.6.3 TUM RGB-D benchmark

The TUM RGB-D dataset collection (Sturm et al. [2012]) contains indoor sequences captured using a Microsoft Kinect v.1 sensor with VGA resolution along with ground truth trajectories of the sensor and a set of tools for easily evaluating the quality of the estimated trajectories. The proposed methods are evaluated on almost all the sequences in the dataset except for those in which scenes are beyond the range of the sensor. The main purpose is to demonstrate the advantage of the proposed ANNF and ONNF over the classical EDF in terms of accuracy and robustness. Since one of the state-of-the-art implementations (Kuse and Shen [2016])



(a) *freiburg2_xyz.* (b) *freiburg3_nostructure_texture_near.*

Figure 5.9: Semi-dense reconstruction of two sequences from the TUM RGB-D benchmark datasets.

terminates the optimization on the QVGA resolution, its results are not as good. To achieve a fair comparison, we implement our own EDF based tracker which outperforms (Kuse and Shen [2016]). Besides, to comprehensively assess the performance, a sparse feature based solution ORB-SLAM2 (RGB-D version) (Mur-Artal and Tardós [2017]) is included in the evaluation. Note however that we only use the tracker of (Mur-Artal and Tardós [2017]) in order to fairly asses pure tracking performance (by setting *mbOnlyTracking=true*) in the experiment.

As shown in Tables 5.3 and 5.4, the ANNF based paradigm achieves better accuracy than EDF (which we attribute to the use of the signed point-to-tangent distance), and ONNF based tracking significantly outperforms other methods due to bias-free estimation. Since edge alignment methods rely on accurate Canny edge detections, it is not surprising to see (Mur-Artal and Tardós [2017]) performs better on several sequences in *feiburg 1*, in which significant image blur due to aggressive rotations occurs. This problem would be less apparent if using a more advanced device, *e.g.* Kinect V2, which is equipped with a global shutter RGB camera. Large RMSEs of edge alignment based methods are also witnessed in other sequences such as *fr3_str_tex_near*, which is caused by an ambiguous structure. Only one edge is detected in the conjunction of two planes with homogeneous color, which notably leads to a tracking failure, as at least one degree of freedom of the motion simply becomes unobservable[4]. In general, however, ANNF and ONNF based trackers work outstandingly well, since the median errors remain reasonably small. To conclude, semi-dense reconstruction results for the sequences *fr2_xyz* and *fr3_nostructure_texture* are given in Fig. 5.9. Since no global optimization is performed, the crispness of these reconstructions again underlines the quality of the edge alignment.

[4]Note furthermore that—in order to achieve a fair comparison—all methods are evaluated up until the same frame if one of the methods loses tracking. Since all methods are equally affectable by failure situations, this does not give preference to any of the methods.

Table 5.3: Relative Pose RMSE(**R**:deg/s, **t**:m/s) of TUM datasets

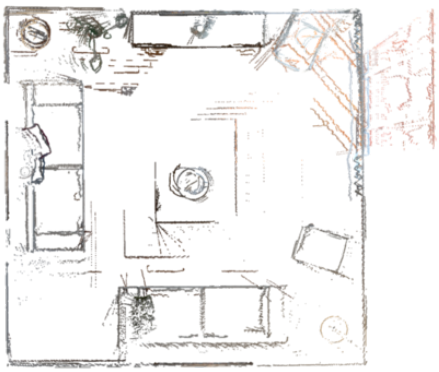
	ORB-SLAM2 Features		Our Implementation (EDF) Edge Alignment		Our Method (ANNF) Edge Alignment		Our Method (ONNF) Edge Alignment	
Seq.	RMSE(R)	RMSE(t)	RMSE(R)	RMSE(t)	RMSE(R)	RMSE(t)	RMSE(R)	RMSE(t)
fr1_360	**2.593**	**0.065**	15.922	0.448	6.450	0.211	4.036	0.121
fr1_desk	2.393	0.051	2.966	0.073	6.312	0.075	**1.923**	**0.031**
fr1_desk2	**3.533**	**0.074**	6.109	0.359	6.513	0.156	5.056	0.131
fr1_floor	2.739	0.038	1.248	0.037	0.880	0.013	**0.823**	**0.010**
fr1_plant	1.837	0.044	**1.486**	**0.028**	2.864	0.050	1.535	0.036
fr1_room	2.721	0.076	2.762	0.084	4.445	0.223	**2.003**	**0.042**
fr1_rpy	2.393	0.037	5.430	0.119	5.699	0.063	**2.245**	**0.034**
fr1_teddy	**2.061**	**0.062**	4.193	0.153	9.248	0.196	2.921	0.123
fr1_xyz	**0.958**	**0.014**	1.178	0.024	1.534	0.045	1.127	0.019
fr2_360_hemisphere	3.482	0.280	7.817	0.833	2.751	0.380	**1.092**	**0.108**
fr2_360_kidnap	2.471	0.174	3.077	0.244	2.151	0.172	**1.161**	**0.084**
fr2_coke	4.845	0.165	4.340	0.102	**1.179**	**0.023**	3.502	0.058
fr2_desk	1.060	0.030	0.463	**0.008**	0.469	**0.008**	**0.458**	**0.008**
fr2_desk_with_person	1.639	0.056	0.472	0.012	**0.462**	**0.008**	0.511	0.009
fr2_dishes	1.624	0.035	0.653	**0.012**	0.915	0.016	**0.629**	**0.012**
fr2_rpy	0.357	0.004	0.321	0.003	**0.318**	**0.003**	0.325	0.004
fr2_xyz	0.328	0.005	0.317	0.004	**0.307**	**0.003**	0.319	0.004
fr3_cabinet	2.976	0.071	2.024	0.040	2.482	0.058	**1.636**	**0.036**
fr3_large_cabinet	**2.369**	**0.100**	4.717	0.214	4.036	0.190	3.278	0.167
fr3_long_office_household	0.906	0.019	0.529	0.011	0.695	0.014	**0.503**	**0.010**
fr3_nostr_tex_far	2.449	0.121	1.306	0.054	9.412	0.522	**0.892**	**0.035**
fr3_nostr_tex_near_withloop	1.591	0.050	7.193	0.164	**1.440**	**0.029**	1.502	0.043
fr3_str_notex_far	**0.453**	**0.013**	1.935	0.132	3.104	0.144	0.588	0.027
fr3_str_notex_near	**3.088**	**0.060**	32.288	0.622	23.482	0.422	25.888	0.752
fr3_str_tex_far	0.618	0.018	0.472	0.013	**0.459**	**0.012**	0.477	0.013
fr3_str_tex_near	0.890	0.017	1.102	0.018	1.167	0.021	**0.593**	**0.010**

Table 5.4: Absolute Trajectory RMSE(m) of TUM datasets

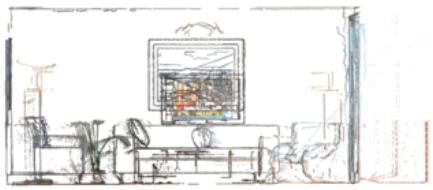
	ORB-SLAM2 Features	Our Implementation (EDT) Edge Alignment	Our Method (ANNF) Edge Alignment	Our Method (ONNF) Edge Alignment
Seq.	RMSE(t)	RMSE(t)	RMSE(t)	RMSE(t)
fr1_360	**0.139**	0.607	0.315	0.242
fr1_desk	0.065	0.168	0.212	**0.044**
fr1_desk2	**0.093**	0.581	0.381	0.187
fr1_floor	0.061	0.019	**0.017**	0.021
fr1_plant	0.067	**0.042**	0.133	0.059
fr1_room	**0.143**	0.248	0.621	0.242
fr1_rpy	0.066	0.109	0.205	**0.047**
fr1_teddy	**0.150**	0.290	0.437	0.193
fr1_xyz	**0.009**	0.053	0.137	0.043
fr2_360_hemisphere	0.213	0.504	0.432	**0.079**
fr2_360_kidnap	0.144	0.249	0.128	**0.122**
fr2_coke	1.521	0.076	**0.029**	0.070
fr2_desk	0.274	0.040	0.039	**0.037**
fr2_desk_with_person	0.135	0.072	**0.047**	0.069
fr2_dishes	0.104	0.035	0.034	**0.033**
fr2_rpy	**0.004**	0.009	0.007	0.007
fr2_xyz	**0.008**	0.009	0.010	**0.008**
fr3_cabinet	0.312	**0.057**	0.103	**0.057**
fr3_large_cabinet	**0.154**	0.351	0.349	0.317
fr3_long_office_household	0.276	0.087	0.090	**0.085**
fr3_nostr_tex_far	0.147	0.055	0.191	**0.026**
fr3_nostr_tex_near_withloop	0.111	0.406	0.101	**0.090**
fr3_str_notex_far	**0.008**	0.157	0.026	0.031
fr3_str_notex_near	**0.091**	0.910	0.813	1.363
fr3_str_tex_far	0.030	0.013	**0.012**	0.013
fr3_str_tex_near	0.045	0.026	0.047	**0.025**

5.6.4 ICL-NUIM Dataset

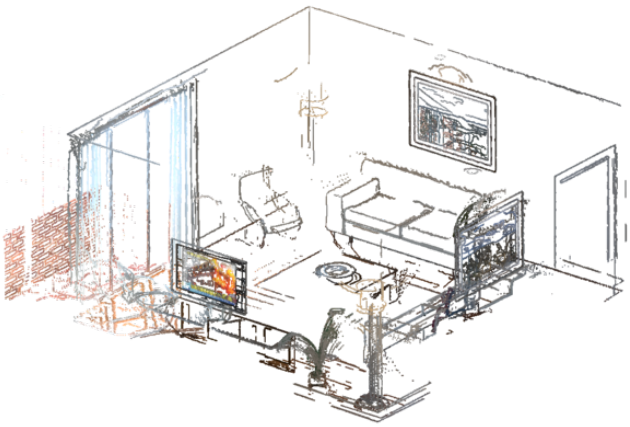
A high-quality indoor dataset for evaluating RGB-D VO/SLAM systems is provided by Handa et al. [2014]. Although it is synthetic, the structure and texture are realistically rendered using a professional 3D content creation software. Illumination and reflection properties are properly taken into account. The algorithm is evaluated using the *living room* collection which contains four sequences composed of different trajectories within the same room. The scene has several challenging elements for VO/SLAM systems, including reflective surfaces, locally texture-poor regions, and multiple illumination sources. The evaluation results are given in Table 5.5 and 5.6. We see that the ONNF based tracker again easily outperforms in the comparison. Since image blur effects do not exist in the synthetic dataset, the advantages of the ONNF based tracking scheme are even more clearly demonstrated. To conclude, we again provide a semi-dense reconstruction of the *living room 2* using ONNF based tracking in Fig. 5.10.



(a) Aerial view of the living room.



(b) Horizontal view of the living room.



(c) A manually selected perspective.

Figure 5.10: Semi-dense reconstruction of the ICL_NUIM *living room* sequence *kt2*.

Table 5.5: Relative pose RMSE **R**:deg/s, **t**:m/s of ICL_NUIM

	ORB-SLAM2 Features		Our Implementation (EDF) Edge Alignment		Our Method (ANNF) Edge Alignment		Our Method (ONNF) Edge Alignment	
Seq.	RMSE(R)	RMSE(t)	RMSE(R)	RMSE(t)	RMSE(R)	RMSE(t)	RMSE(R)	RMSE(t)
living room 0	1.186	0.030	2.717	0.082	1.766	0.047	**0.674**	**0.014**
living room 1	0.464	0.022	0.590	0.030	1.297	0.059	**0.208**	**0.009**
living room 2	2.997	0.103	0.544	0.029	0.307	0.013	**0.269**	**0.011**
living room 3	0.367	0.012	0.214	0.011	0.157	**0.007**	**0.152**	**0.007**

Table 5.6: Absolute Trajectory RMSE (m) of ICL_NUIM

	ORB-SLAM2 Features	Our Implementation (EDF) Edge Alignment	Our Method (ANNF) Edge Alignment	Our Method (ONNF) Edge Alignment
Seq.	RMSE(t)	RMSE(t)	RMSE(t)	RMSE(t)
living room 0	0.043	0.113	0.074	**0.035**
living room 1	0.082	0.080	0.119	**0.023**
living room 2	0.108	0.089	0.038	**0.031**
living room 3	0.015	0.016	**0.008**	**0.008**

5.6.5 ANU-RSISE Sequence

We captured and analyzed our own large-scale indoor RGB-D sequence, a scan of a complete level of the Research School of Engineering at the Australian National University (ANU). It is more challenging than most of the TUM datasets for at least two reasons. First, the scene is an open-space office area of approximately 300 m^2, with highly self-similar locations. A footprint of the building is shown in Fig. 5.11. The illumination is not as consistent as in small-scale environments, such as a desk or a small office room. Second, the sequence contains a combination of challenging structures such as reflecting surfaces (window glass) and cluttered objects (plants). We use the Microsoft Kinect v2 for data collection, and the RGB and depth images are prealigned and resized to VGA resolution, similar to what has been done in the TUM benchmark sequences.

All algorithms are evaluated qualitatively by visualizing the reconstruction results in Fig. 5.12. The global BA module of (Mur-Artal and Tardós [2017]) is again disabled to underline pure tracking performance. Although (Mur-Artal and Tardós [2017]) performs very well along straight parts, severe problems are witnessed in the corners. The bad tracking is due to the reflection imaging on the window glass, which generates false features. All edge alignment based trackers still perform well in the corner taking advantage of good signal-to-noise ratio and the proposed robust weighting strategies. The advantages of the ANNF and ONNF over the EDF are clearly seen over the straight parts. By looking at the two recycle bins (blue and red) near the starting point, ONNF performs the best in terms of start-to-end error. A more detailed map and some close-up shots occurring during the exploration using ONNF based tracking are given in Fig. 5.13.

5.6.6 Efficiency Analysis

Real-time performance is typically required for any VO system in a practical application. To see the improvement in terms of efficiency, we compare the computation time of each method on a desktop with a Core i7-4770 CPU. As seen in Fig. 5.14, the computation in the tracking thread consists of four parts: Canny edge detection (CE), distance transformation (DT), optimization (Opt), and others. As claimed before, the DT computation of the ANNF is almost as fast as the EDF[5], while the ONNF is the most efficient due to the adaptive sampling and the parallel computation. Another significant difference occurs in the optimization. The EDF based method takes more time than the ANNF because of the requirement for bilinear interpolation during the evaluation of the objective function. ONNF based tracking is basically as fast as ANNF based tracking. The difference in the optimization time for nearest neighbour field based approaches is due to another modification that could be equally applied as a speed-up factor to all other fields. We include a stochastic optimization strategy, which starts with a small number of 3D points and gradually increases the amount until reaching the minimum, where optimization over all points is performed. Note that the result in Fig. 5.14 is normalized over the number of points (at most 6500) and it includes the computation on the whole image pyramid (from level 0 to level 2). The keyframe generation thread runs at about 10 Hz in parallel.

[5]The adaptive sampling function is switched off for ANNF in the efficiency analysis.

(a) Floorplan of level 3 of the ANU Research School of Engineering.

(b) Typical snapshots of the environment.

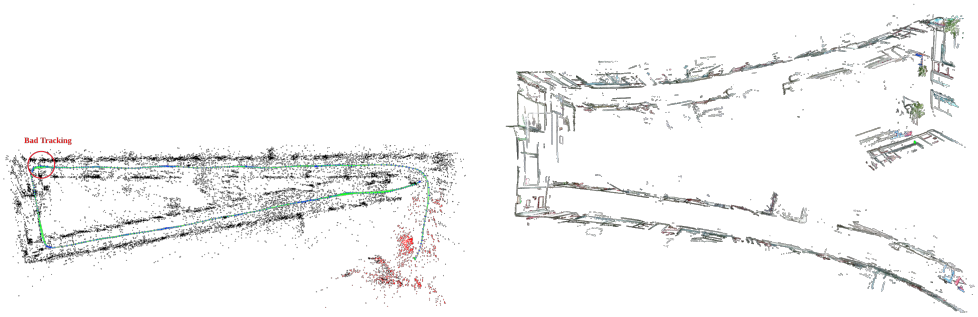
Figure 5.11: The schematic trajectory of the sensor when collecting the sequence is illustrated in (a). The sequence starts from the position highlighted with a green dot. Structures such as window glass, plants, dark corridor caused by inconsistent illumination that make the sequence challenging are shown in (b).

Even using three pyramid levels, our method achieves 20Hz and thus real-time processing on a standard CPU. The main bottleneck in the computation is the image processing. Considering that this could be offloaded into embedded hardware, we believe that our method represents an interesting choice that ultimately could make semi-dense processing accessible to computationally constrained devices.

5.7 Conclusion

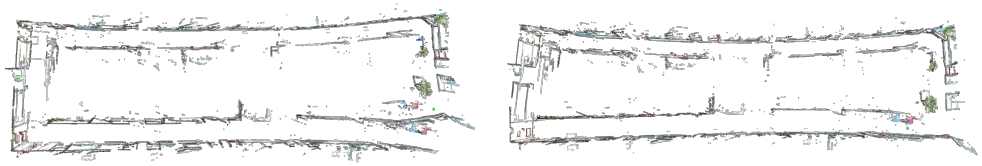
The chapter introduces approximate nearest neighbour fields as a valid, at least equally accurate alternative to euclidean distance fields in 3D-2D curve alignment with clear benefits in computational efficiency. We furthermore prove that the bias plaguing distance field based registration in the case of partially observed models is effectively encountered through an orientation of the nearest neighbour fields, thus re-establishing the model-to-data registration paradigm as the most efficient choice for geometric 3D-2D curve alignment. We furthermore prove that efficient sub-sampling strategies are readily accessible to nearest neighbour field extraction.

The geometric approach to semi-dense feature-based alignment has the clear advantages of resilience to illumination changes and the ability to be included in a curve-based bundle adjustment that relies on a global, spline-based representation of the structure. With a focus on the efficient formulation of residual errors in curve alignment, we believe that the present investigation represents an important addition to this line of research.



(a) Rconstruction result by ORB-SLAM2 (RGB-D version with local BA enabled).

(b) Reconstruction result by smoothed EDF based tracker.



(c) Reconstruction result by ANNF based tracker.

(d) Reconstruction result by ONNF based tracker

Figure 5.12: Evaluation on our own indoor sequence. The figures show different perspectives of the result obtained with and without loop closure enabled.

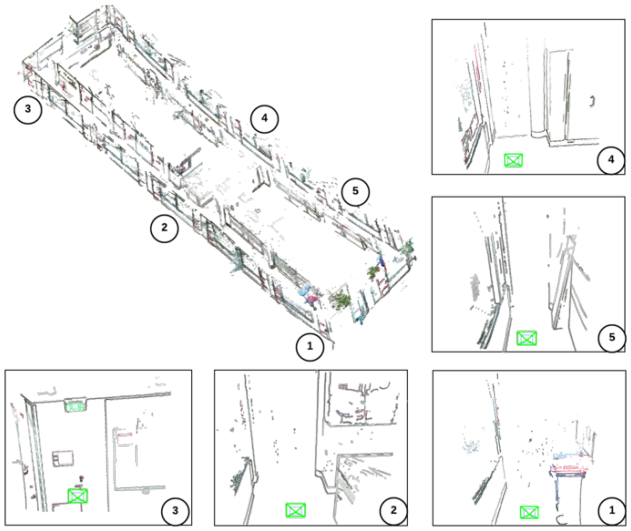


Figure 5.13: Close-up perspectives during the exploration of level 3 of the ANU Research School of Engineering.

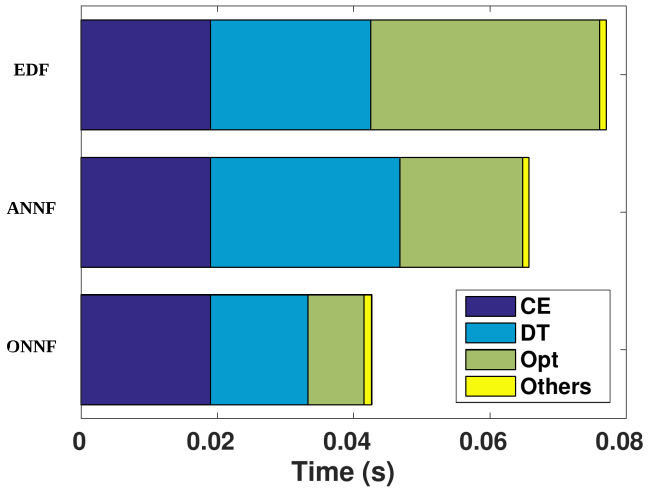


Figure 5.14: Efficiency analysis on EDF, ANNF and ONNF based tracker.

Chapter 6

Semi-Dense 3D Reconstruction with a Stereo Event Camera

Compared to the proposed pipelines that take advantage of piece-wise planar and Manhattan World environments, the edge alignment based method proposed in Chapter. 5 expands the range of exploration by utilizing the more general structural regularity — edges. However, in many practical robotic applications, exiting pipelines including the proposed ones in the previous chapters may come across challenges such as aggressive motion, high dynamic range, *etc.* . These challenging scenarios typically go beyond the limitation of standard cameras. Thus, a new camera architecture is investigated in this chapter, and a novel 3D reconstruction method is developed.

Event cameras are bio-inspired sensors that offer several advantages such as low latency, high-speed and high dynamic range to tackle challenging scenarios in computer vision. This chapter presents a solution to the problem of 3D reconstruction from data captured by a stereo event-camera rig moving in a static scene, such as in the context of stereo Simultaneous Localization and Mapping. The proposed method consists of the optimization of an energy function designed to exploit narrow-baseline spatio-temporal consistency of events triggered across both stereo image planes. To improve the density of the reconstruction as well as to reduce the uncertainty of the estimation, a probabilistic depth fusion strategy is also developed. The resulting method has no special requirements on either the motion of the stereo event-camera rig or on prior knowledge about the scene. Experiments demonstrate our method can deal with both texture-rich scenes as well as sparse scenes, outperforming state-of-the-art stereo methods based on event data image representations.

6.1 Introduction

Event cameras, such as the Dynamic Vision Sensor (DVS) (Lichtsteiner et al. [2008]), are novel devices that report pixel-wise intensity changes (called "events") asynchronously, at the time they occur. As opposed to standard cameras, they do not acquire an entire image frame at the same time nor do they operate at a fixed frame rate. This asynchronous and differential principle of operation reduces power and bandwidth requirements drastically. Endowed with microsecond temporal resolution, event cameras are able to capture high-speed motions, which would typically cause severe motion blur with standard cameras. In addition, event cameras

have a very high dynamic range (HDR) (*e.g.* 140 dB compared to 60 dB of most standard cameras), which allows them to be used under broad illuminations. Hence, event cameras open the door to tackle challenging scenarios that are inaccessible to standard cameras, such as high-speed and/or HDR tracking (Mueggler et al. [2014]; Lagorce et al. [2015]; Zhu et al. [2017]), control (Conradt et al. [2009]; Delbruck and Lang [2013]) and Simultaneous Localization and Mapping (SLAM) (Kim et al. [2016]; Rebecq et al. [2017c]).

The main challenge in visual processing with event cameras is to devise specialized algorithms that can exploit the temporally asynchronous and spatially sparse nature of the image data produced by DVS cameras, hence unlock their full potentials, whereas existing computer vision algorithms designed for conventional cameras do not directly apply in general. Some preliminary works on DVS addressed this issue by combining event cameras with other sensors, such as standard cameras (Censi and Scaramuzza [2014]; Kueng et al. [2016]) or depth sensors (Censi and Scaramuzza [2014]; Weikersdorfer et al. [2014]), in order to simplify the task at hand. Although this approach obtained certain success, the true potential of an event camera has not been fully exploited since parts of such combined systems are limited by the lower dynamic range devices. In this work, we tackle the problem of stereo 3D reconstruction for visual odometry (VO) or SLAM using event cameras alone. Our goal is to unlock the potential of event cameras by developing a method that directly works on the DVS principles using raw DVS signals.

6.1.1 Related work on Event-based Depth Estimation

The majority of works on depth estimation with event cameras target the problem of "instantaneous" stereo, i.e., 3D reconstruction using events from a pair of synchronized cameras in stereo configuration (i.e., with a fixed baseline), during a very short time (ideally on a per-event basis). Some of these works (Kogler et al. [2011]; Rogister et al. [2012]; Camunas-Mesa et al. [2014]) follow the classical paradigm of solving stereo in two steps: epipolar matching followed by 3D point triangulation. Temporal coherence (e.g., simultaneity) of events across both left and right cameras is used to find matching events, and then standard triangulation (Hartley and Zisserman [2003]) recovers depth. Other works, such as (Piatkowska et al. [2013]), extend cooperative stereo (Marr and Poggio [1976]) to the case of event cameras. These methods are typically demonstrated in scenes with static cameras and few moving objects, so that event matches are easy to find due to uncluttered event data.

Works (Schraml et al. [2016, 2015]) also target the problem of "instantaneous" stereo (depth maps produced using events over very short time intervals), but they use two non-simultaneous event cameras. These methods exploit a constrained hardware setup (two rotating event cameras with known motion) to either (*i*) recover intensity images on which conventional stereo is applied (Schraml et al. [2016]) or (*ii*) match events across cameras using temporal metrics and then use triangulation (Schraml et al. [2015]).

Recently, depth estimation with a single event camera has been shown in (Rebecq et al. [2016]; Kim et al. [2016]; Rebecq et al. [2017a]). These methods recover a semi-dense 3D reconstruction of the scene by integrating information from the events of a moving camera over a longer time interval, and therefore, require information of the relative pose between the camera and the scene. Hence, these methods do not target the problem of "instantaneous"

depth estimation but rather the problem of depth estimation for visual odometry and SLAM.

6.1.2 Contribution

This chapter, the problem of 3D reconstruction (in the SLAM context) with a pair of event cameras in stereo configuration is addressed. The proposed approach is based on temporal coherence of events across left and right image planes. However, it differs from previous efforts, such as the "instantaneous" stereo methods (Kogler et al. [2011]; Rogister et al. [2012]; Camunas-Mesa et al. [2014]; Schraml et al. [2016, 2015]), in that: (*i*) we do not follow the classical paradigm of event matching plus triangulation, but rather a forward-projection approach that allows us to estimate depth without explicitly solving the event matching problem, (*ii*) we are able to handle sparse scenes (events generated by few moving objects) as well as cluttered scenes (events constantly generated everywhere in the image plane due to the motion of the camera), and (*iii*) we use camera pose information to integrate observations over time to produce semi-dense depth maps. Moreover, our method computes continuous depth values, as opposed to other methods, such as (Rebecq et al. [2016]), which discretize the depth range. Finally, in contrast to monocular methods that target the same depth-for-SLAM problem, we are able to recover the absolute scale of the scene.

Outline. Section 6.2 presents the 3D reconstruction problem considered and our solution, formulated as the minimization of an objective function that measures the temporal inconsistency of event history maps across left and right image planes. Section 6.3 presents an approach to fuse multiple event-based 3D reconstructions into a single depth map. Section 6.4 evaluates our method on both synthetic and real event data, showing the good performance of our method.

6.2 3D Reconstruction by Event Time-History Maps Energy Minimization

Our method is inspired by multi-view stereo pipelines for conventional cameras, such as DTAM (Newcombe et al. [2011b]), which aim at maximizing the photometric consistency through a number of narrow-baseline video frames. However, since event cameras do not output absolute intensity but rather intensity changes (the "events"), the direct photometric-consistency-based method cannot be applied readily. Instead, we exploit the fact that event cameras encode visual information in the form of microsecond-resolution timestamps of intensity changes.

For a stereo event camera, a detectable[1] 3D point in the overlapping field of view (FOV) of the cameras will generate an event on both left and right cameras. Ideally, these two events should spike at the exact same time and their coordinates should be corresponding to each other defined by the epipolar geometry between the two cameras in stereo configuration. This

[1] A point at an intensity edge (i.e., non-homogeneous region of space), so that intensity changes (i.e., events) are generated when the point moves relative to the camera.

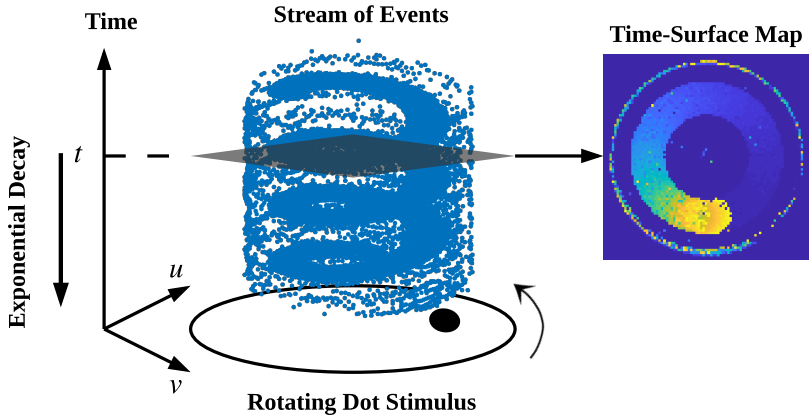


Figure 6.1: Left: output of an event camera when viewing a rotating dot. Right: Time-surface map (6.1) at a time t, $\mathcal{T}(\mathbf{x},t)$, which essentially measures how far in time (with respect to t) the last event spiked at each pixel $\mathbf{x} = (u,v)^T$. The brighter the color, the more recently the event was generated.

property actually enables us to apply (and modify) an idea similar to DTAM, simply by replacing the photometric consistency with the stereo temporal consistency. However, as shown in (Benosman et al. [2011]), the stereo temporal consistency does not strictly hold at the pixel level because of signal latency and jitter effects. Thus, we define our stereo temporal consistency by aggregating measurements over spatio-temporal neighborhoods, rather than by comparing the event timestamps at two individual pixels. Technical details will be given in the sequel.

6.2.1 Event Time-History Maps

We propose to apply patch-match to compare a pair of spike-history maps, in place of the photometric warping error as used in the conventional DTAM method. Specifically, to create two distinctive maps, we advocate the use of *Time-Surface* inspired by (Lagorce et al. [2016]) for event-camera based pattern recognition. As illustrated in Fig. 6.1, the output of an event camera is a stream of temporal events, where each event $e_k = (u_k, v_k, t_k, p_k)$ consists of the space-time coordinates where the intensity change of predefined size happened and the sign (polarity $p_k \in \{+1,-1\}$) of the change[2]. The time-surface map at time t is defined by applying an exponential decay kernel on the last spiking time t_{last} at each pixel coordinate $\mathbf{x} = (u,v)^T$:

$$\mathcal{T}(\mathbf{x},t) \doteq \exp\left(-\frac{t - t_{\text{last}}(\mathbf{x})}{\delta}\right), \tag{6.1}$$

where δ, the decay rate parameter, is a small constant number (*e.g.* , $30ms$ in our experiments). For convenient visualization and processing, (6.1) is further rescaled to the range $[0,255]$. Our objective function is constructed on a set of time-surface maps (6.1) at different observation times $t = \{t_s\}$.

[2]Event polarity is not used, as Rebecq et al. [2017a] show that it is not needed for 3D reconstruction.

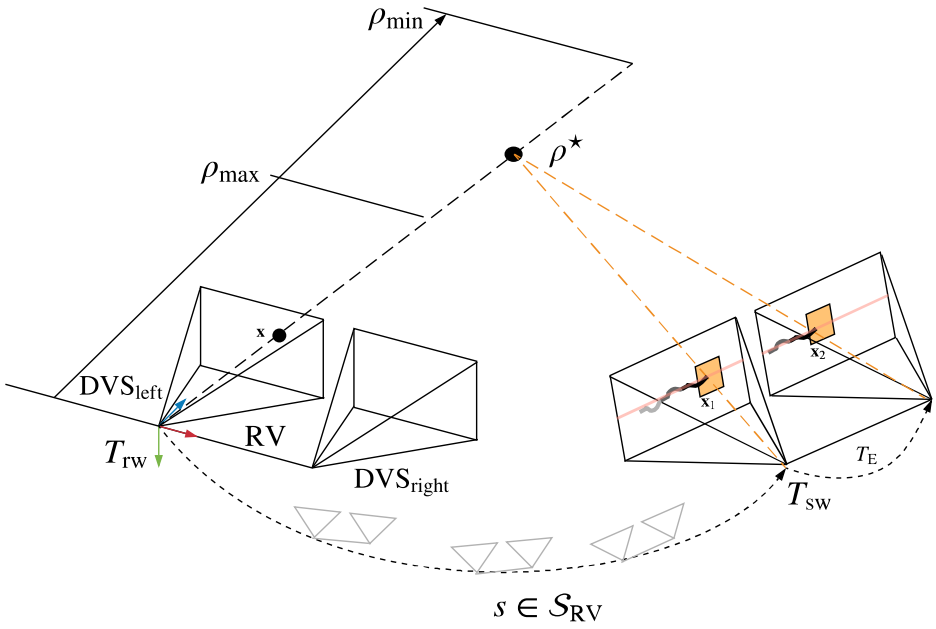


Figure 6.2: Illustration of the geometry of the proposed problem and solution. The reference view (RV) is on the left, in which an event with coordinates $\mathbf{x}$ is back-projected into 3D space with a hypothetical inverse depth ρ. The optimal inverse depth $\rho^\star$, lying inside the search interval $[\rho_{\min}, \rho_{\max}]$, corresponds to the real location of the 3D point which fulfills the temporal consistency in each neighbouring stereo observation s.

6.2.2 Problem Formulation

We follow a global energy minimization framework to estimate the inverse depth map $\mathcal{D}$ in the reference view (RV) from a number of stereo observations $s \in \mathcal{S}_{\text{RV}}$ nearby. A *stereo observation* at time t refers to a pair of time-surface maps created using (6.1), $\big(\mathcal{T}_{\text{left}}(\cdot,t), \mathcal{T}_{\text{right}}(\cdot,t)\big)$. A stereo observation could be triggered by either a pose update or at a constant rate. For each pixel $\mathbf{x}$ in the reference view, its inverse depth $\rho^\star \doteq 1/z^\star$ is estimated by optimizing the objective function:

$$\begin{aligned} \rho^\star &= \arg\min_{\rho} C(\mathbf{x},\rho), \\ C(\mathbf{x},\rho) &\doteq \frac{1}{|\mathcal{S}_{\text{RV}}|} \sum_{s\in\mathcal{S}_{\text{RV}}} \| \tau^s_{\text{left}}(\mathbf{x}_1(\rho)) - \tau^s_{\text{right}}(\mathbf{x}_2(\rho)) \|_2^2, \end{aligned} \tag{6.2}$$

where $|\mathcal{S}_{\text{RV}}|$ denotes the number of involved neighboring stereo observations, which is used for averaging. The function $\tau^s_{\text{left/right}}(\mathbf{x})$ returns the temporal information $\mathcal{T}_{\text{left/right}}(\cdot,t)$ inside a $w \times w$ patch centered at image point $\mathbf{x}$. The residual,

$$r_s(\rho) \doteq \| \tau^s_{\text{left}}(\mathbf{x}_1(\rho)) - \tau^s_{\text{right}}(\mathbf{x}_2(\rho)) \|_2, \tag{6.3}$$

denotes the temporal difference in l_2 norm between patches centered at $\mathbf{x}_1$ and $\mathbf{x}_2$ in the left and right event cameras, respectively.

The geometry behind the proposed objective function is illustrated in Fig. 6.2. Since

we assume the calibration (intrinsic and extrinsic parameters) as well as the pose of the left event camera at each observation are known, the points $\mathbf{x}_1$ and $\mathbf{x}_2$ are given by $\mathbf{x}_1(\rho) = \pi(T_{sr}\pi^{-1}(\mathbf{x},\rho))$ and $\mathbf{x}_2(\rho) = \pi(T_{\mathrm{E}}T_{sr}\pi^{-1}(\mathbf{x},\rho))$, respectively, where function $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ projects a 3D point onto the camera's image plane, while its inverse function $\pi^{-1} : \mathbb{R}^2 \to \mathbb{R}^3$ back-projects a pixel into 3D space given the inverse depth ρ, and T_{E} denotes the transformation from the left to the right event camera. Note that all event coordinates $\mathbf{x}$ are undistorted and rectified.

To verify that the proposed objective function does lead to the optimum depth for a generic event in the reference view (Fig. 6.3(a)), a number of stereo observations from a real stereo event-camera sequence (Zhu et al. [2018]) have been created (Figs. 6.3(c) and 6.3(d)), and are used to visualize the overall energy at the event location (Fig. 6.3(b)). The size of the patch is $w = 25pixel$ in this work.

Observe that our approach significantly departs from classical two-step event-processing methods (Kogler et al. [2011]; Rogister et al. [2012]; Camunas-Mesa et al. [2014]) that solve the stereo matching problem first and then triangulate the 3D point, which is prone to errors due to the difficulty in establishing correct event matches during very short time intervals. These two-step approaches work in a "back-projection" fashion, mapping 2D event measurements to 3D space. Instead, our approach combines matching and triangulation in a single step, operating in a forward-projection manner (from 3D space to 2D event measurements). As shown in Fig. 6.2, an inverse depth hypothesis ρ yields a 3D point, $\pi^{-1}(\mathbf{x},\rho)$, whose projections on both stereo image planes for all times "s" gives curves $\mathbf{x}_1^s(\rho)$ and $\mathbf{x}_2^s(\rho)$ that are compared in the objective function (6.2). Hence, an inverse depth hypothesis ρ establishes candidate stereo event matches, and the best matches are obtained once the objective function has been minimized with respect to ρ.

6.2.3 Inverse Depth Estimation

The proposed objective function (6.2) is optimized using non-linear least squares methods. The Gauss-Newton method is used here, which iteratively discovers the root of the necessary optimality condition

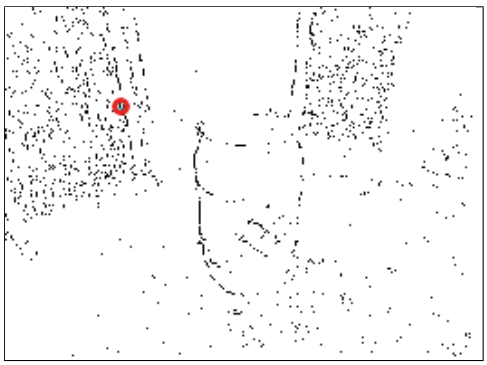
$$\frac{\partial C}{\partial \rho} = \frac{2}{|\mathcal{S}_{\mathrm{RV}}|}\sum_{s\in\mathcal{S}_{\mathrm{RV}}} r_s \frac{\partial r_s}{\partial \rho} = 0. \tag{6.4}$$

Substituting the linearization of r_s at ρ_k using the first order Taylor formula, $r_s(\rho_k + \Delta\rho) \approx r_s(\rho_k) + J_s(\rho_k)\Delta\rho$, in (6.4) we obtain

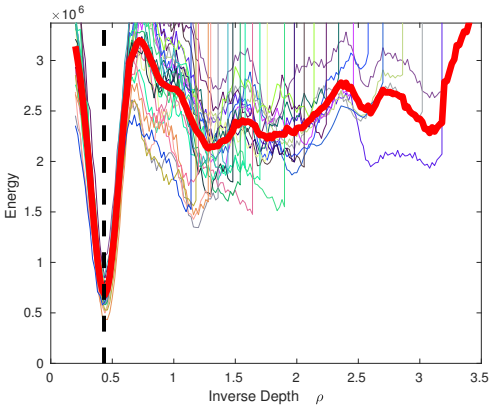
$$\sum_{s\in\mathcal{S}_{\mathrm{RV}}} J_s(r_s + J_s\Delta\rho) = 0, \tag{6.5}$$

where both, residual $r_s \equiv r_s(\rho_k)$ and Jacobian $J_s \equiv J_s(\rho_k)$, are scalars. Consequently the inverse depth ρ is iteratively updated by adding the increment

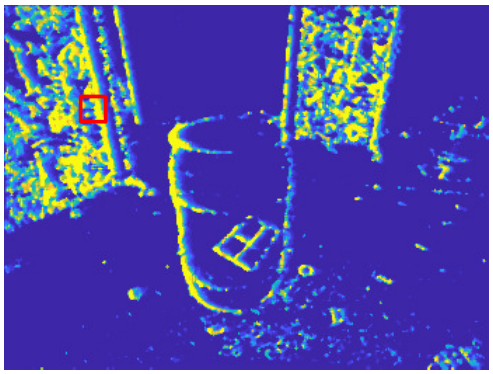
$$\Delta\rho = -\frac{\sum_{s\in\mathcal{S}_{\mathrm{RV}}} J_s r_s}{\sum_{s\in\mathcal{S}_{\mathrm{RV}}} J_s^2}. \tag{6.6}$$



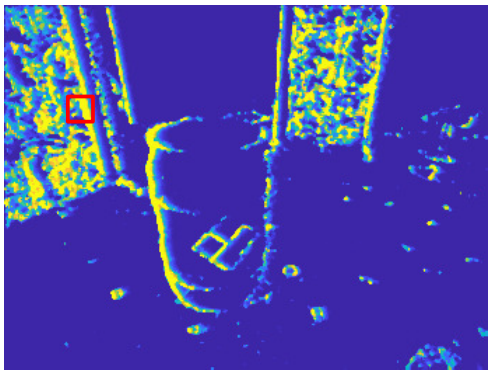
(a) Events in the reference view (RV).



(b) Objective function (6.2) (in red).



(c) Time-Surface Map (left DVS).



(d) Time-Surface Map (right DVS).

Figure 6.3: Verification of the proposed objective function. A randomly selected event in the reference view (RV) is marked by a red circle in (a). The overall energy is visualized in (b), with a red curve obtained by averaging the cost of all valid neighbouring observations (indicated by curves with random colors). The vertical dashed line (black) indicates the groundtruth inverse depth. The time-surface map of the left and the right event cameras at one of the observation times are shown in (c) and (d), respectively, where the patches for measuring the temporal residual are marked by red rectangles.

The Jacobian is computed by applying the chain rule,

$$\begin{aligned} J_s(\rho) &\doteq \frac{\partial}{\partial \rho} \left\| \tau^s_{\text{left}}(\mathbf{x}_1(\rho)) - \tau^s_{\text{right}}(\mathbf{x}_2(\rho)) \right\|_2 \\ &= \frac{1}{\left\| \tau^s_{\text{left}} - \tau^s_{\text{right}} \right\|_2 + \epsilon} \left(\tau^s_{\text{left}} - \tau^s_{\text{right}} \right)^T_{1\times w^2} \left(\frac{\partial \tau^s_{\text{left}}}{\partial \rho} - \frac{\partial \tau^s_{\text{right}}}{\partial \rho} \right)_{w^2 \times 1}, \end{aligned} \tag{6.7}$$

where, for simplicity, the pixel $\mathbf{x}_i(\rho)$ is omitted in the last equation. To avoid division by zero, a small number ϵ is added to the length of the residual vector. Actually, as shown by an investigation on the distribution of the temporal residual r_s in Section 6.3.1, the temporal residual is unlikely to be close to zero for valid stereo observations (*i.e.* , patches with enough

Algorithm 3 Inverse Depth Estimation at a Reference View (RV)

Input: pixel $\mathbf{x}$, stereo event observations $\mathcal{T}^s_{\text{left}}, \mathcal{T}^s_{\text{right}}$ and poses T_{sr}, T_{E}.
$\rho_0 \leftarrow \rho_{\text{initial}}$ (by coarse search over a range $[\rho_{\min}, \rho_{\max}]$).
while not converged **do**
    for each observation s **do**
        Compute $r_s(\rho_k)$ in (6.3).
        Compute $J_s(\rho_k)$ using (6.7).
    end for
    Update: $\rho_k \leftarrow \rho_k + \Delta\rho$, using (6.6).
end while
return Inverse depth ρ_k.

events occurred). The derivative of the time-surface map with respect to the inverse depth is calculated by

$$\frac{\partial \boldsymbol{\tau}^s}{\partial \rho} = \frac{\partial \boldsymbol{\tau}^s}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \rho} = \left(\frac{\partial \boldsymbol{\tau}^s}{\partial u}, \frac{\partial \boldsymbol{\tau}^s}{\partial v} \right)_{w^2 \times 2} \left(\frac{\partial u}{\partial \rho}, \frac{\partial v}{\partial \rho} \right)^T. \tag{6.8}$$

The computation of $\frac{\partial u}{\partial \rho}$ and $\frac{\partial v}{\partial \rho}$ are given in Appendix. A.2.1.

The overall procedure is summarized in Algorithm 3. The inputs of the algorithm are, respectively, the pixel coordinate $\mathbf{x}$ of an event in the RV, a set of stereo observations (time-surface maps) $\mathcal{T}^s_{\text{left/right}}$ ($s \in \mathcal{S}_{\text{RV}}$), the relative pose T_{sr} from the RV to each involved stereo observation s and the constant extrinsic parameters between both event cameras, T_{E}. The inverse depths of all events in the RV are estimated independently. Therefore, the computation is parallelisable. The convergence basin is first localized by a coarse search over the range of plausible inverse depth values followed by a nonlinear refinement using the Gauss-Newton method.

6.3 Semi-Dense Reconstruction

The 3D reconstruction method presented in Section 6.2 produces a sparse depth map at the reference view (RV). To improve the density of the reconstruction while reducing the uncertainty of the estimated depth, we run the reconstruction method (Algorithm 3) on several RVs along time and fuse the results. To this end, the uncertainty of the inverse depth estimation is studied in this section. Based on the derived uncertainty, a fusion strategy is developed and is incrementally applied as sparse reconstructions of new RVs are obtained.

6.3.1 Uncertainty of Inverse Depth Estimation

In the last iteration of Gauss-Newton's method, the inverse depth is updated by

$$\rho^\star \equiv \rho_k \leftarrow \rho_k + \Delta\rho(\mathbf{r}), \tag{6.9}$$

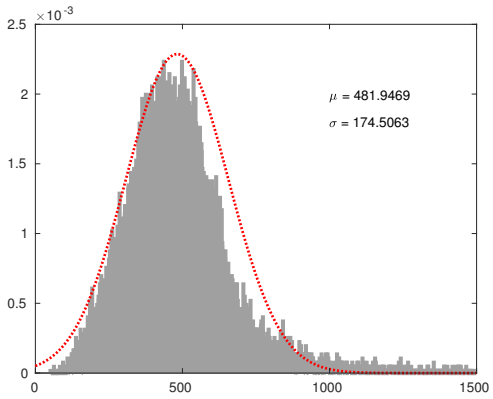


Figure 6.4: Distribution of the temporal residuals and Gaussian fit $\mathcal{N}(\mu, \sigma^2)$.

where $\Delta\rho$ is a function of the residuals $\mathbf{r} \doteq \{r_1, r_2, \ldots, r_s \,|\, s \in \mathcal{S}_{\text{RV}}\}$ as defined in (6.6). The variance $\sigma^2_{\rho^\star}$ of the inverse depth estimate can be derived using uncertainty propagation (Mur-Artal and Tardos [2015]). For simplicity, only the noise in the temporal residuals $\mathbf{r}$ is considered:

$$\sigma^2_{\rho^\star} \approx \left(\frac{\partial \rho^\star}{\partial \mathbf{r}}\right)^T \begin{pmatrix} \sigma_r^2 & & & \\ & \sigma_r^2 & & \\ & & \ddots & \\ & & & \sigma_r^2 \end{pmatrix} \frac{\partial \rho^\star}{\partial \mathbf{r}} = \frac{\sigma_r^2}{\sum_{s \in \mathcal{S}_{\text{RV}}} J_s^2}. \tag{6.10}$$

The derivation of this equation can be found in Appendix. A.2.2. Instead of assigning σ_r arbitrarily, we determine it empirically by investigating the distribution of the temporal residuals $\mathbf{r}$. Using the ground truth depth, we sample a large number of temporal residuals $\mathbf{r} = \{r_1, r_2, ..., r_n\}$. The variance σ_r^2 is obtained by fitting a Gaussian distribution to the histogram of $\mathbf{r}$, as illustrated in Fig. 6.4.

6.3.2 Inverse Depth Fusion

In order to improve the density of the reconstruction, inverse depth estimates of multiple RVs are incrementally transferred to a selected reference view, RV*, and fused. Assuming the inverse depth of a pixel in RV_i follows a distribution $\mathcal{N}(\rho_a, \sigma_a^2)$, its corresponding location in RV* is typically a non-integer pixel coordinate $\mathbf{x}^f$, which will have an effect on the four neighbouring pixel coordinates $\mathbf{x}_1^i, \mathbf{x}_2^i, \mathbf{x}_3^i, \mathbf{x}_4^i$. Using $\mathbf{x}_1^i$ as an example, the fusion is performed based on the following rules:

1. Assign $\mathcal{N}(\rho_a, \sigma_a^2)$ to $\mathbf{x}_1^i$ if no previous distribution exists.

2. If there is an existing inverse depth distribution assigned at $\mathbf{x}_1^i$, *e.g.* , $\mathcal{N}(\rho_b, \sigma_b^2)$, the compatibility between the two inverse depth hypotheses is checked to decide whether they are fused. The compatibility is evaluated by using the χ^2 test at 95% (Mur-Artal and Tardos [2015]):

$$\frac{(\rho_a - \rho_b)^2}{\sigma_a^2} + \frac{(\rho_a - \rho_b)^2}{\sigma_b^2} < 5.99. \tag{6.11}$$

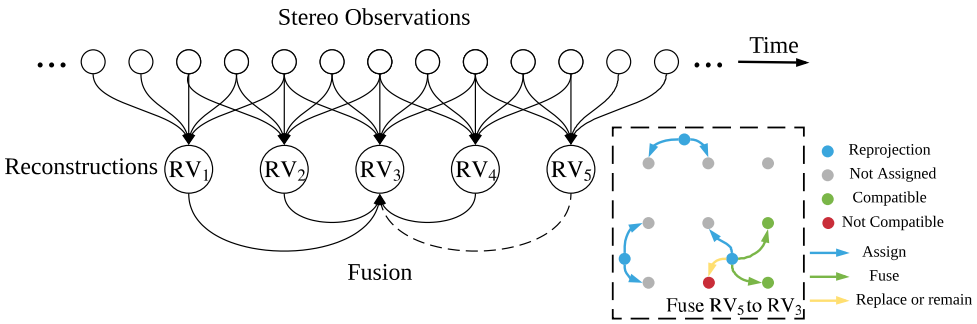


Figure 6.5: Illustration of the fusion strategy. All stereo observations $(\mathcal{T}^s_{\text{left}}, \mathcal{T}^s_{\text{right}})$ are denoted by hollow circles and listed in chronological order. Neighbouring RVs are fused into a chosen RV (*e.g.* , RV_3). Using the fusion from RV_5 to RV_3 as an example, the fusion rules are illustrated in the dashed square, in which a part of the image plane is visualized. The blue dots are the reprojections of 3D points in RV_5 on the image plane of RV_3. Gray dots represent unassigned pixels which will be assigned by blue dots within one pixel away. Pixels that have been assigned, *e.g.* the green ones (compatible with the blue ones) will be fused. Pixels that are not compatible (in red) will either remain or be replaced, depending on which distribution has the smaller uncertainty.

If the two hypotheses are compatible, they are fused into a single inverse depth distribution by:

$$\mathcal{N}\left(\frac{\sigma_a^2\rho_b + \sigma_b^2\rho_a}{\sigma_a^2 + \sigma_b^2}, \frac{\sigma_a^2\sigma_b^2}{\sigma_a^2 + \sigma_a^2}\right), \tag{6.12}$$

otherwise the distribution with a smaller variance remains.

An illustration of the fusion strategy is given in Fig. 6.5

6.4 Experiment

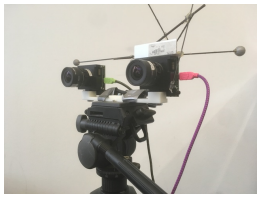
The proposed stereo 3D reconstruction method is evaluated in this section. We first introduce the configuration of our stereo event-camera system and the information of the datasets used in the experiments. Afterwards, both quantitative and qualitative evaluations are given. Additionally, the depth fusion process is illustrated to give an impression on how it improves the density of the reconstruction while reducing depth uncertainty.

6.4.1 Stereo Event-camera Setup

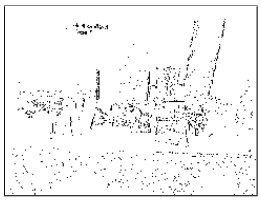
To evaluate our method, we use sequences from publicly available simulators (Mueggler et al. [2017]) and datasets (Zhu et al. [2018]), and we also collect our own sequences using a stereo event-camera rig (Fig. 6.6). The stereo rig consists of two Dynamic and Active Pixel Vision Sensors (DAVIS) (Brandli et al. [2014]) of $240 \times 180 pixel$ resolution, which are calibrated intrinsically and extrinsically using *Kalibr* (Furgale et al. [2013]). Since our algorithm is working

Camera:	DAVIS240 ×2
Width:	240
Height:	180
AFOV:	62.9°
Baseline:	14.7 cm

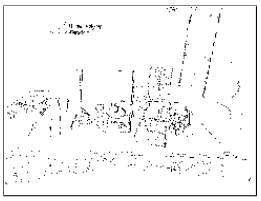
(a) System information.



(b) Stereo event-camera rig.



(c) Event map on the left camera.



(d) Event map on the right camera.

Figure 6.6: Left, (a) and (b): the stereo event-camera rig used in our experiment, consisting of two synchronized DAVIS (Brandli et al. [2014]) devices. Right, (c) and (d): rectified event maps at one time observation.

on rectified and undistorted coordinates, the joint undistortion and rectification transformation are computed in advance.

As the stereo event-camera system moves, a new stereo observation $(\mathcal{T}^s_{\text{left}}, \mathcal{T}^s_{\text{right}})$ is generated when a pose update is available. The generation consists of two steps. The first step is to generate a rectified event map by collecting all events that occurred within $10ms$ (from the pose's updating time to the past), as shown in Fig. 6.3(a). The second step is to refresh the time-surface maps in both left and right event cameras, as shown in Figs. 6.3(c) and 6.3(d). One of the observations is selected as the RV. The rectified event map of the RV together with the rest of the observations are fed to the inverse depth estimation module (Algorithm 3). We use the rectified event map as a selection map, i.e. we estimate depth values only at the pixels with non zero values in the rectified event map (as shown in Fig. 6.6 (c) and (d)). As more and more RVs are reconstructed and fused together, the result becomes both more dense and more accurate.

6.4.2 Results

The evaluation is performed on six sequences, including a synthetic sequence from the simulator (Mueggler et al. [2017]), three sequences collected by ourselves (hand-held) and two sequences from Zhu et al. [2018] (with a stereo event camera mounted on a drone). A snapshot of each scene is given in the first column of Fig. 6.7. In the synthetic sequence, the stereo event-camera system looks orthogonally towards three frontal parallel planes while performing a pure translation. Our three sequences showcase typical office scenes with various office supplies. The stereo event-camera rig is hand-held and performs arbitrary 6-DOF motion, which is recorded by a motion-capture system. The other two sequences are collected in a large indoor environment using a drone (Zhu et al. [2018]), with pose information also from a motion-capture system. These two sequences are very challenging for two reasons: (*i*) a wide variety of structures such as chairs, barrels, a tripod on a cabinet, *etc.* can be found in this scene, and (*ii*) the drone undergoes relatively high-speed motions during data collection.

Quantitative evaluation on datasets with groundtruth depth are given in Table 6.1, where we compare our method with two state-of-the-art instantaneous stereo matching methods, "Fast Cost-Volume Filtering" (FCVF) (Hosni et al. [2013]) and "Semi-Global Matching" (SGM) (Hirschmuller [2008]), working on pairs of time-surface images. We report the *mean* depth

Table 6.1: Quantitative evaluation on sequences with groundtruth depth.

	Dataset	*simulation_3planes* (Mueggler et al. [2017])	*Indoor_flying1* (Zhu et al. [2018])	*Indoor_flying3* (Zhu et al. [2018])
	Depth range	2.76 m	4.96 m	5.74 m
Our Method	Mean error	**0.03** m	**0.13** m	**0.33** m
	Median error	**0.01** m	**0.05** m	**0.11** m
	Relative error	**1.17** %	**2.65** %	**5.79** %
FCVF (Hosni et al. [2013])	Mean error	0.05 m	0.99 m	1.03 m
	Median error	0.03 m	0.25 m	**0.11** m
	Relative error	1.84 %	20.8 %	17.3 %
SGM (Hirschmuller [2008])	Mean error	0.08 m	0.93 m	1.19 m
	Median error	0.03 m	0.31 m	0.20 m
	Relative error	3.22 %	18.7 %	20.8 %

error, the *median* depth error and the relative error (defined as the *mean* depth error divided by the depth range of the scene (Rebecq et al. [2017a])). In fairness to the comparison, the fully dense depth maps returned by FCVF and SGM are masked by the non-zero pixels in the time-surface images. Besides, the boundary of the depth maps are cropped considering the block size used in each implementation. The best results per sequence are highlighted in bold in Table 6.1. Our method outperforms the other two competitors on all sequences. Although FCVF and SGM also give satisfactory results on the synthetic sequence, they do not work well in more complicated scenarios in which the observations are either not dense enough, or the temporal consistency does not strictly hold in a single stereo observation.

Reconstruction results on all sequences are visualized in Fig. 6.7. Images on the first column are raw intensity frames from the DAVIS. They convey the appearance of the scenes but are not used by our algorithm. The second column shows rectified and undistorted event maps in the left event-camera of a RV. The number of the events depends on not only on the motion of the stereo rig but also on the amount of visual contrast in the scene. Semi-dense depth maps (after fusion with several neighbouring RVs) are given in the third column, in which hot colors refer to close while cold colors mean far. The last column visualizes the 3D point cloud of each sequence at a chosen perspective. Note that only points whose uncertainty (σ_ρ) are smaller than $0.8 \times \sigma_\rho^{\max}$ are visualized in 3D.

The reconstruction of the rectified events in one RV is sparse and full of noise typically. To show how the fusion strategy improves the density of the reconstruction as well as reduces the uncertainty, we additionally perform an experiment that visualizes the fusion process incrementally. As shown in Fig. 6.8, the first column visualizes the uncertainty maps before the fusion. The second to the fourth column demonstrates the uncertainty maps after fusing the result of a RV with its neighbouring 4, 8 and 16 estimations, respectively. Hot colors refer to high uncertainty while cold colors mean low uncertainty. The result becomes increasingly dense and accurate as more and more RVs are fused. Note that the remaining highly uncertain estimates generally correspond to events that are caused by either noise or low-contrast patterns.

6.5 Conclusion

This chapter has proposed a novel and effective solution to 3D reconstruction using a pair of temporally-synchronized event cameras in stereo configuration. This is, to the best of the author's knowledge, the first one to address such a problem allowing stereo SLAM applications with event cameras. The proposed energy minimization method exploits spatio-temporal

consistency of the events across cameras to achieve high accuracy (between 1% and 5% relative error in depth), and it outperforms state-of-the-art stereo methods using the same spatio-temporal image representation of the event stream. Future work includes the development of a full stereo visual odometry system, by combining the proposed 3D reconstruction strategy with a stereo-camera pose tracker.

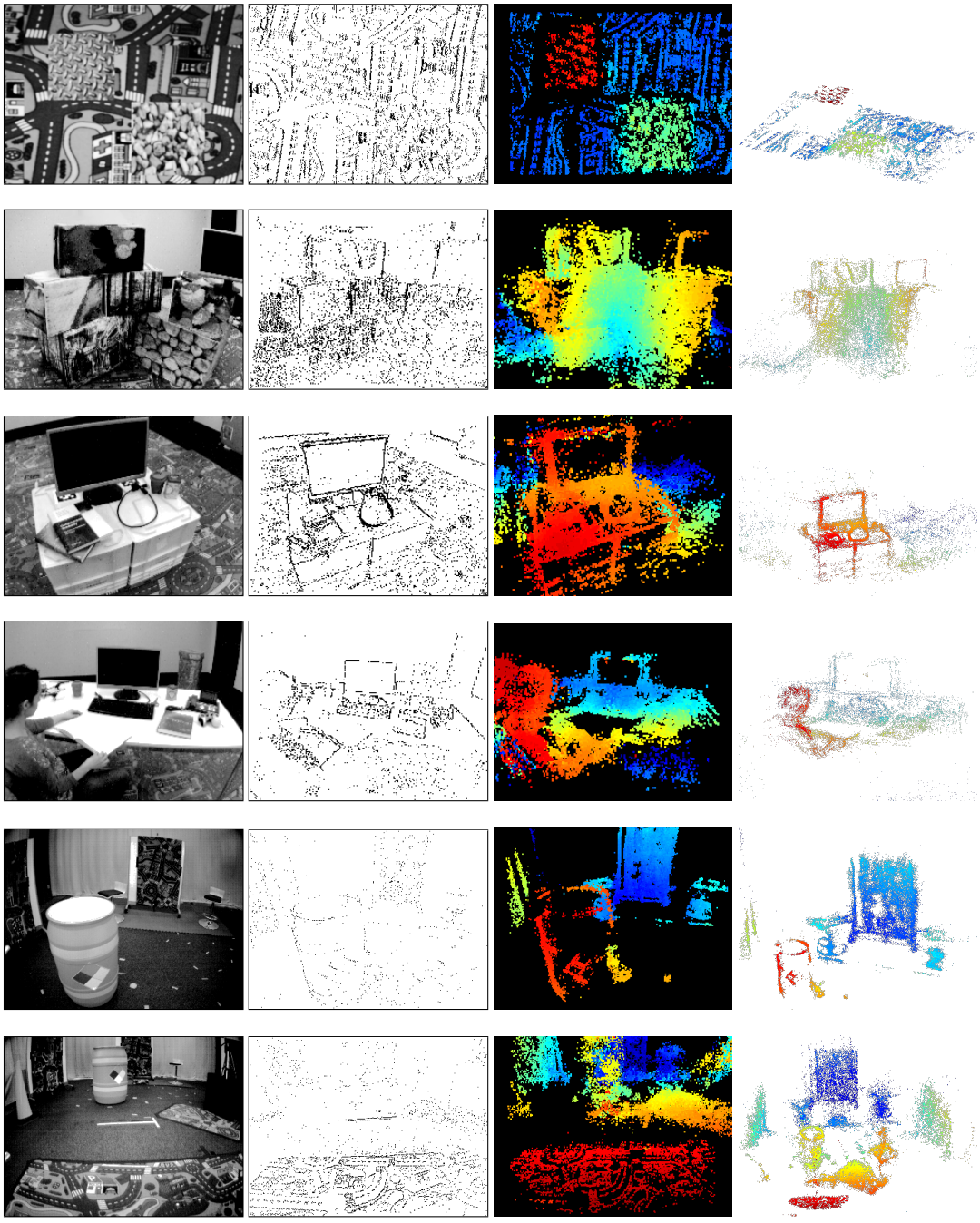


Figure 6.7: Results of the proposed method on several datasets. Images on the first column are raw intensity frames (not rectified nor lens-distortion corrected). The second column shows the events (undistorted and rectified) in the left event camera of a reference view (RV). Semi-dense depth maps (after fusion with several neighbouring RVs) are given in the third column, colored according to depth, from red (close) to blue (far). The fourth column visualizes the 3D point cloud of each sequence at a chosen perspective. No post-processing, such as regularization through median filtering (Rebecq et al. [2017a]), was performed.

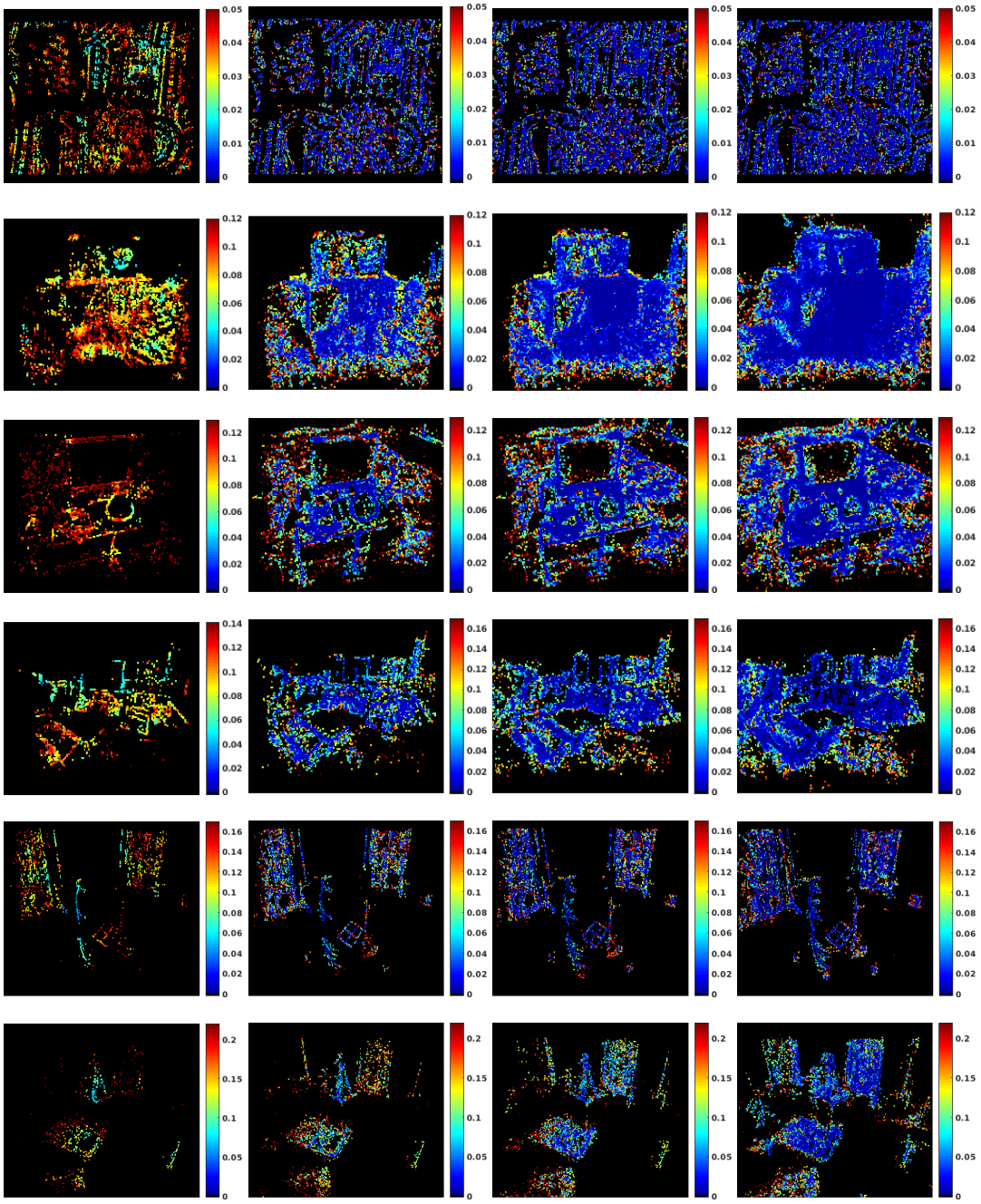


Figure 6.8: Illustration of how the fusion strategy increasingly improves the density of the reconstruction while reducing depth uncertainty. The first column shows the uncertainty maps σ_ρ before the fusion. The second to the fourth columns report the uncertainty maps after fusing with 4, 8 and 16 neighbouring estimations, respectively.

Chapter 7

Summary and Future Work

Visual Odometry remains one of the fundamental research topics in robotic vision. As its application domain becomes larger, we witness new challenging scenarios where application-specific conditions occur. This drives us to develop more accurate, efficient and robust systems based on new, tailormade theories and smart implementations.

7.1 Summary and Contributions

This dissertation is devoted to developing high-performance vision based motion estimation pipelines by exploiting the structural regularity in man-made environments. The results push the limits of the state-of-the-art in various aspects, such as efficiency, accuracy and robustness. Beyond that, to be prepared for more challenging scenarios in high-speed robotic applications, we investigate the event-based camera — a sensor that recently attracts huge attention in the community. A summary of the contributions is given in the following.

7.1.1 Improving Efficiency, Accuracy and Robustness

To answer the old question — how to determine the fundamental matrix from planes -, we have presented a novel two-step linear method, which demonstrates that the compatibility equation can be used for calculating the fundamental matrix in a linear way. Even in the presence of noise in the homographies, the proposed method can provide numerically stable and accurate results.

In piece-wise planar environments, we have presented an efficient method of rotation estimation for depth cameras in Chapter 3. It tracks planar modes from the distribution of surface normal vectors and estimates the relative rotation by registering the bundle of planar modes in the current frame to that of the reference frame. Based on this idea, a full 6 DoF motion estimation pipeline is developed in Chapter 4 for depth sensors in Manhattan Worlds. Thanks to the spatial regularity in Manhattan Worlds, the estimations of the rotation and translation can be decoupled, even for each degree of freedom in the translation. The derived camera orientation is furthermore absolute and thus free of long-term drift. Our method relies on the alignment of density distribution functions, a concept which has linear complexity in the number of points. We achieve not only competitive accuracy but also superior computational efficiency at the same time. The algorithm runs at 50 Hz on a laptop with a CPU. Live demos can be found on

Youtube [1] and an exemplary implementation is open sourced on my Github repository [2].

In Chapter 5, two novel distance transform alternatives to the classical Euclidean distance fields are presented. They bring clear benefits to 3D-2D curve alignment in terms of accuracy and efficiency. The bias plaguing distance filed based registration in the case of partially observed models is proved to be effectively encountered through an orientation of the nearest neighbour fields, thus re-establishing the model-to-data registration paradigm as the most efficient choice for geometric 3D-2D curve alignment. Furthermore, an adaptive sampling strategy is employed to the computation of nearest neighbour fields, which significantly boosts the computational performance. Additionally, to improve the robustness of the motion estimation, the registration is formulated as a maximum a posteriori problem. The statistical characteristics of the sensor model are learned to suppress noise and outliers.

7.1.2 Exploration of Novel Camera Architectures

Event-based cameras provide a few outstanding properties, such as low latency, high dynamic range and low bandwidth requirement. These advantages are key to dealing with challenging scenarios in practical robotic applications. The main challenge in visual processing with event cameras is to devise specialized algorithms that can exploit the temporally asynchronous and spatially sparse nature of the image data produced by DVS cameras, hence unlock their full potential. In chapter 6 we focus on the problem of event-based local mapping — one of the main challenges of event-based VO systems, and present a pipeline of semi-dense 3D reconstruction with a stereo event camera. To the best of the author's knowledge, the proposed method is the first work to address this problem allowing stereo SLAM applications with event cameras. A video regarding this work can be found on Youtube [3] and an exemplary implementation is open sourced on my Github repository [4].

It is worth recording the real inspiration to this work (not discussed in the regarding publication to avoid misleading readers) when I read the work by Benosman et al. [2011], although not focusing on the depth estimation. In fact, it demonstrates how the epipolar geometry is discovered from events and their co-activation sets. The paper shows that the temporal correlation is hard to be verified by a single observation because of outliers originating from non-negligible effects of latency and jitter. Therefore, they introduce a new concept called co-activation probability field, which is created by accumulating the occurrence of temporal neighbors when the stereo DVS is placed at different distances away from a planar monitor. The epipolar line is consequently determined by a set of activated models and the fundamental matrix is able to be worked out. The success of this method is attributed to the theory of Hebb observation, which states "*If the inputs to a system cause the same pattern of activity to occur repeatedly, the set of active elements constituting that pattern will become increasingly strongly inter-associated*" (Hebb [2005]). Following this logic, in the case of event-based depth estimation, stereo observations are the inputs to the system (*i.e.* the "pattern" determining the depth of an event) and the outputs (*i.e.* "active elements constituting the pattern") refer to some activities that are

[1] https://www.youtube.com/watch?v=W06ZZki0rTA&t=5s
[2] https://github.com/Ethan-Zhou/MWO
[3] https://www.youtube.com/watch?v=Qrnpj2FD1e4&t=3s
[4] https://github.com/Ethan-Zhou/SDR-ECCV-2018-

consistent with a uniform model under a latent constraint. The task of solving the event-based depth estimation turns to exploiting a proper constraint among multiple stereo observations. This lateral thinking ultimately led me to try a strategy of the event-based multi-view stereo, rather than an instantaneous method.

7.2 Future Work

The proposed methods are certainly possible to be further refined. Discussions on the potential improvements have already been provided in the conclusion part of each chapter. Three particularly meaningful extensions to the proposed systems are discussed in the following.

7.2.1 Towards an Agile and Robust VO System for RGB-D Cameras

VO systems that use standard RGB(-D) cameras suffer from blurry images when sudden motion occurs. To deal with this problem, the most mature strategy is to fuse vision and inertial information. An inertial measurement unit (IMU) provides noisy but high-frequent and outlier-free measurements for translational acceleration and angular velocity. Due to the upper bound on the random walk as a function of the integration time, IMUs can provide relatively accurate position and orientation changes over short integration periods. In fact, the combination of any vision system with an IMU is believed to be able to provide reliable tracking of even aggressive motion due to the complementary nature of these sensors. At the beginning of the extension work, we plan to implement a loosely-coupled method, which would fuse pre-integrated IMU measurements and the relative pose measurements from the Canny-VO. Besides, we will simultaneously work on accelerating the registration part of Canny-VO. Since our proposed nearest neighbour fields report the coordinates of closest points, the idea of EPnP (Lepetit et al. [2009]) is possibly applicable. By updating the pose via a closed-form solution during each iteration, this PnP based solution will make the registration much faster than the non-linear optimization.

Due to the limited sensing range of an RGB-D camera, the current Canny-VO system is restricted to indoor environments. When changing to outdoor environments, the RGB-D are only occasionally valid, thus no longer supporting the introduced 3D-2D registration approach. This leads to a tracking failure. Since a VO system typically does not have a re-localization module, it has to be re-initialized which will almost inevitably lead to severe drift in form of a jump in the pose. In order to make the Canny-VO system work outdoors, the mapping thread must be able to reconstruct edges that are out of the sensing range of the depth camera. To estimate the depth of edge pixels, a multi-view stereo based method can be implemented. For an edge pixel (with unknown depth) in a keyframe, the method tracks its location along the epipolar line. At each neighbouring perspective (with known relative pose), a depth estimate can be obtained through triangulation and as presented in (Engel et al. [2013]), the respective uncertainty may also be derived. The final depth estimation of each edge pixel is obtained by fusing all compatible estimates together. A regularization operation may be required to enhance the smoothness of the depth estimates along edges.

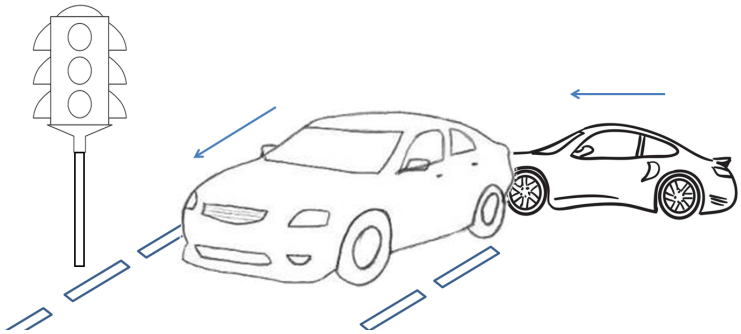


Figure 7.1: Illustration of the dynamic (piece-wise rigid) case. Three independent motions exist in the scene: two cars having uncorrelated motion and the camera's motion with respect to the static background (*e.g.* the traffic light and the traffic lines).

7.2.2 Detection and Tracking of Independent Motions Using 3D Edges

The ability to work in dynamic scenes is a cornerstone for any SLAM/VO systems. Although it is possible by using pure visual information as shown by Tan et al. [2013], a more robust solution, from the perspective of engineering, is given by adding additional sensors such as an IMU (Lynen et al. [2015]). This type of device senses the ego-motion of the platform, and that is exactly why it increases the system's ability to distinguish static background features from the rest. However, the ability of estimating ego motion even in a dynamic environment is not enough for a machine that claims to be able to perceive its surroundings. Perhaps equally important ability is to recognize independently moving objects and what exact motion (piece-wise rigid) they are performing. This ability of dynamic perception is of utmost importance in autonomous driving scenario where an intelligent vehicle needs to make decisions not only based on its own dynamic state in the environment, but also based on the dynamic state of other vehicles. An exemplary scene is illustrated in Fig.7.1.

Following the idea of the proposed Canny-VO system, we model the environment with 3D curves. Given two RGB-D frames I_1 and I_2 as the reference frame and the current frame, respectively, the goal is to detect and estimate all piece-wise rigid movements of subsets of the 3D edges. To achieve the goal, we need to solve two sub problems: motion segmentation and motion estimation. Motion segmentation refers to clustering 3D curves (or their 2D projections inside a reference frame) into different groups according to the rigid motion they comply with. Motion estimation consists of computing the relative motion of the independent moving objects with respect to the camera using 3D-2D edge alignment. The overall problem is a classical chicken and egg problem, as finding the optimal motions requires having identified all the segments that belong to its cluster, and finding all the segments requires knowing the motion. Inspired by a optical flow work (Yang and Li [2015]) that uses piecewise parametric model, the overall problem is formulated as a multi-model fitting problem, which can be solved through a scheme of discrete-continuous optimization. The discrete part is a combinatorial optimization problem where each curve segment is assigned with a label, while the continuous part consists of updating the label pool (namely the motion models that each label refers to). A potential formulation of this idea is given in the following.

7.2.2.1 Energy function

Let $\mathcal{L} = \{1, \ldots, K\}$ be a set of discrete labels representing the set of motion models, denoted as $\Theta = \{\boldsymbol{\theta}_k\}$,[5] $k = 1, \ldots, K$. Let $\mathcal{P}_E$ be the 3D edge map domain, Ω_E be the corresponding 2D edge pixel domain, and $L : \mathcal{P}_E \rightarrow \mathcal{L}$ be a labelling function. Assigning label k to a 3D edge point $\mathbf{p}$ in the reference frame means that the reprojection of $\mathbf{p}$ to the current frame is determined by motion $\boldsymbol{\theta}_k \in \Theta$. The overall energy function is defined as,

$$E(\Theta, L) = E_D(\Theta, L) + \lambda_P E_P(L) + \lambda_C E_C(L), \tag{7.1}$$

which consists of a data term $E_D(\Theta, L)$, an inter-curve compatibility term $E_C(L)$ and a Potts model term $E_P(L)$.

- Data term

$$E_D(\Theta, L) = \sum_{\mathbf{x} \in \Omega_E} r(\pi(T(\boldsymbol{\theta}_{L(\mathbf{p})}, \mathbf{p}))) = \sum_{\mathbf{p} \in \mathcal{P}_E} r(\boldsymbol{\theta}_{L(\mathbf{p})}, \mathbf{p}) \tag{7.2}$$

  where $\mathbf{p} := d(\mathbf{x})\pi^{-1}(\mathbf{x})$, $T(\cdot, \cdot)$ transforms $\mathbf{p}$ to the current frame with the given motion $\boldsymbol{\theta}_{L(\mathbf{p})}$, $\pi(\cdot)$ is the world-to-camera function projecting a 3D point onto the image plane, and $r(\cdot)$ is the geometric residual error returned by a distance field (defined in the current frame). For simplicity, $T(\cdot, \cdot)$ and $\pi(\cdot)$ are dropped off in Eq. 7.2. The data term assesses how well an edge point complies with a motion model geometrically.

- Potts model term
  To encourage spatially coherent labelling, a pairwise Potts model is used. The term is defined on the discrete labelling variables as,

$$E_P(L) = \sum_{\mathbf{x}, \mathbf{x}' \in \mathcal{E}_E} \delta(L(\mathbf{p}) - L(\mathbf{p}')) \tag{7.3}$$

  where $\delta(\cdot)$ is the 0-1 indicator function which takes 1 if the input argument is true, and 0 otherwise.

- Inter-edge compatibility term
  When two edges complying with different motion intersect at a point in the image plane (*e.g.* a "T" conjunction), the Potts model tends to unify the labelling for better spatial coherence. To have a fine labelling result, we introduce a term that does not penalize the variations (even if they may be large) between neighbouring pixels within a single edge segment. It only penalizes motion discontinuities at edge intersections. Let $\mathcal{E}_E$ denotes two connected edge pixels, the inter-edge compatibility term is defined as,

$$E_C(L) = \sum_{\mathbf{x}, \mathbf{x}' \in \mathcal{E}_E} \rho(|r(\boldsymbol{\theta}_{L(\mathbf{p})}, \bar{\mathbf{p}}) - r(\boldsymbol{\theta}_{L(\mathbf{p}')}, \bar{\mathbf{p}})|), \tag{7.4}$$

  where $\bar{\mathbf{p}} = d(\bar{\mathbf{x}})\pi^{-1}(\bar{\mathbf{x}})$, $\bar{\mathbf{x}}$ denotes the midpoint of $(\mathbf{x}, \mathbf{x}')$.

[5] $\boldsymbol{\theta} := \{\mathbf{R}, \mathbf{t}\}$ represents the motion parameters.

7.2.2.2 Alternated Optimization

- **Labeling: Solve for L with fixed Θ**
  With a fixed motion set Θ, the energy minimization reduces to a labeling problem with energy
  $$E(L) = E_D(L) + \lambda_P E_P(L) + \lambda_C E_C(L). \tag{7.5}$$
  The energy is a standard Markov Random Filed (MRF). The $\alpha-$expansion based graph-cut method Boykov et al. [2001] can be used for fast approximate energy minimization.

- **Fitting: Solve for Θ with fixed L**
  The motion Θ appears only in data term E_D. With a fixed label set L, the energy minimization turns to a number of registration problems as we perform in the Chapter 5. Each rigid motion $\boldsymbol{\theta}_k$ could be estimated independently by optimizing
  $$E(\boldsymbol{\theta}_k) = E_D(\boldsymbol{\theta}_k) = \sum_{\mathbf{p}\in\mathcal{P}_E} r(\boldsymbol{\theta}_k, \mathbf{p}). \tag{7.6}$$

7.2.3 Semi-Dense Visual Odometry Using a Stereo Event Camera

The work discussed in Chapter 6 only provides an event-based mapping solution. In order to create a full event-based VO system, a tracker is needed. Since the output of our mapper is a 3D semi-dense map with no appearance information, a tracker that is based on the 3D-2D geometric distance minimization is applicable. Actually, this tracker is exactly what we used in Chapter 5. The good news is that the event-based camera does not suffer from image blur. Thus, a sharp edge map can be obtained on the current frame through an integration of events. In the bootstraping step, an instantaneous stereo matching method (*e.g.* Hirschmuller [2008]) is applied on a pair of time-surface map, which returns a coarse depth map to the tracker. The mapper will update the depth map incrementally using new observations with estimated poses. As we have seen in the results of chapter 6, the quality of the semi-dense depth map is limited by the spatial resolution of the event-based camera. Thus, it would be ideal to follow up on this line of research by using the latest release of the DVS sensor, which supports VGA resolution (Son et al. [2017]).

It would be interesting to test the developed VO system in practical cases such as a traffic scene. Traffic scenes at night consist of many challenging elements for classic VO pipelines using standard cameras, such as 1) inconsistent illumination (high dynamic range), 2) high speed motion of the cameras and 3) independently moving objects (especially those that cannot be observed clearly by standard cameras), *etc.* Due to the asynchronous nature and high temporal resolution, event camera based VO solutions are able to deal with the first two issues. However, the third issue needs additional efforts to deal with. A potential idea is inspired by video stabilization, which estimates the principal flow pattern that can compensate the ego motion of the camera. Since we use only event cameras, the stabilization will be performed on the time-surface map. After applying the flow on the time-surface map, events that are caused by the ego motion of the event camera will be suppressed, whilst the portion of events that are induced by independent moving objects are still spiking. This operation actually provides a way to detect and filter independent motion, even if the independent motion is very fast. Thus,

the static assumption can be guaranteed much better, which in turn improve the accuracy of the motion estimation.

Appendix

A.1 Derivations in Regards to Robust Geometric 3D-2D Edge Alignment

A.1.1 Derivation on Jacobian Matrix of ANNF based Tracking

The linearization of the residual function at $\boldsymbol{\theta}_k$ is

$$r_{lin,i}(\boldsymbol{\theta}_{k+1}) = r_i(\boldsymbol{\theta}_k) + \mathbf{J}_i(\boldsymbol{\theta}_k)\Delta\boldsymbol{\theta}. \tag{A.1}$$

The Jacobian matrix could be obtained using chains rule as

$$\mathbf{J}_i(\boldsymbol{\theta}_k) = g(\mathbf{x}_i)^T \mathbf{J}_\pi \mathbf{J}_T \mathbf{J}_G. \tag{A.2}$$

Each sub Jacobian matrix are derived as following.

$$\mathbf{J}_\pi = \frac{\partial \pi}{\partial T}|_{\mathbf{p}=T(G(\boldsymbol{\theta}_k),\mathbf{x}_i)} = \begin{bmatrix} f_x \frac{1}{z'} & 0 & -f_x \frac{x'}{z'^2} \\ 0 & f_y \frac{1}{z'} & -f_y \frac{y'}{z'^2} \end{bmatrix}, \tag{A.3}$$

where $\mathbf{p}'_i = (x', y', z')$ is the 3D point transformed by motion $G(\boldsymbol{\theta}_k)$.

$$\begin{aligned} \mathbf{J}_T &= \frac{\partial T}{\partial G}|_{G=G(\boldsymbol{\theta}_k),\mathbf{p}=\mathbf{p}_i} \\ &= \begin{bmatrix} x & 0 & 0 & y & 0 & 0 & z & 0 & 0 & 1 & 0 & 0 \\ 0 & x & 0 & 0 & y & 0 & 0 & z & 0 & 0 & 1 & 0 \\ 0 & 0 & x & 0 & 0 & y & 0 & 0 & z & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{A.4}$$

$\mathbf{J}_G$ can be obtained by computing the derivatives of the pose G with respect to the motion parameter $\boldsymbol{\theta} = [t_1, t_2, t_3, c_1, c_2, c_3]^T$, shown as below

$$\mathbf{J}_G = \begin{bmatrix} \frac{\partial r_{11}}{\partial t_1} & \frac{\partial r_{11}}{\partial t_2} & \frac{\partial r_{11}}{\partial t_3} & \frac{\partial r_{11}}{\partial c_1} & \frac{\partial r_{11}}{\partial c_2} & \frac{\partial r_{11}}{\partial c_3} \\ \frac{\partial r_{21}}{\partial t_1} & \frac{\partial r_{21}}{\partial t_2} & \frac{\partial r_{21}}{\partial t_3} & \frac{\partial r_{21}}{\partial c_1} & \frac{\partial r_{21}}{\partial c_2} & \frac{\partial r_{21}}{\partial c_3} \\ \frac{\partial r_{31}}{\partial t_1} & \frac{\partial r_{31}}{\partial t_2} & \frac{\partial r_{31}}{\partial t_3} & \frac{\partial r_{31}}{\partial c_1} & \frac{\partial r_{31}}{\partial c_2} & \frac{\partial r_{31}}{\partial c_3} \\ \frac{\partial r_{12}}{\partial t_1} & \frac{\partial r_{12}}{\partial t_2} & \frac{\partial r_{12}}{\partial t_3} & \frac{\partial r_{12}}{\partial c_1} & \frac{\partial r_{12}}{\partial c_2} & \frac{\partial r_{12}}{\partial c_3} \\ \frac{\partial r_{22}}{\partial t_1} & \frac{\partial r_{22}}{\partial t_2} & \frac{\partial r_{22}}{\partial t_3} & \frac{\partial r_{22}}{\partial c_1} & \frac{\partial r_{22}}{\partial c_2} & \frac{\partial r_{22}}{\partial c_3} \\ \frac{\partial r_{32}}{\partial t_1} & \frac{\partial r_{32}}{\partial t_2} & \frac{\partial r_{32}}{\partial t_3} & \frac{\partial r_{32}}{\partial c_1} & \frac{\partial r_{32}}{\partial c_2} & \frac{\partial r_{32}}{\partial c_3} \\ \frac{\partial r_{13}}{\partial t_1} & \frac{\partial r_{13}}{\partial t_2} & \frac{\partial r_{13}}{\partial t_3} & \frac{\partial r_{13}}{\partial c_1} & \frac{\partial r_{13}}{\partial c_2} & \frac{\partial r_{13}}{\partial c_3} \\ \frac{\partial r_{23}}{\partial t_1} & \frac{\partial r_{23}}{\partial t_2} & \frac{\partial r_{23}}{\partial t_3} & \frac{\partial r_{23}}{\partial c_1} & \frac{\partial r_{23}}{\partial c_2} & \frac{\partial r_{23}}{\partial c_3} \\ \frac{\partial r_{33}}{\partial t_1} & \frac{\partial r_{33}}{\partial t_2} & \frac{\partial r_{33}}{\partial t_3} & \frac{\partial r_{33}}{\partial c_1} & \frac{\partial r_{33}}{\partial c_2} & \frac{\partial r_{33}}{\partial c_3} \\ \frac{\partial t_1}{\partial t_1} & \frac{\partial t_1}{\partial t_2} & \frac{\partial t_1}{\partial t_3} & \frac{\partial t_1}{\partial c_1} & \frac{\partial t_1}{\partial c_2} & \frac{\partial t_1}{\partial c_3} \\ \frac{\partial t_2}{\partial t_1} & \frac{\partial t_2}{\partial t_2} & \frac{\partial t_2}{\partial t_3} & \frac{\partial t_2}{\partial c_1} & \frac{\partial t_2}{\partial c_2} & \frac{\partial t_2}{\partial c_3} \\ \frac{\partial t_3}{\partial t_1} & \frac{\partial t_3}{\partial t_2} & \frac{\partial t_3}{\partial t_3} & \frac{\partial t_3}{\partial c_1} & \frac{\partial t_3}{\partial c_2} & \frac{\partial t_3}{\partial c_3} \end{bmatrix}_{12\times 6} \tag{A.5}$$

$$= \begin{bmatrix} \mathbf{O}_3 & \mathbf{A}_1 \\ \mathbf{O}_3 & \mathbf{A}_2 \\ \mathbf{O}_3 & \mathbf{A}_3 \\ \mathbf{I}_3 & \mathbf{O}_{3\times 3} \end{bmatrix}$$

let's denote $K = 1 + c_1^2 + c_2^2 + c_3^3$, then the entries of the matrices $\mathbf{A}_1$ are,

- $\frac{\partial r_{11}}{\partial c_1} = \frac{2c_1}{K} - \frac{2c_1(1+c_1^2-c_2^2-c_3^2)}{K^2}$
- $\frac{\partial r_{11}}{\partial c_2} = -\frac{2c_2}{K} - \frac{2c_2(1+c_1^2-c_2^2-c_3^2)}{K^2}$
- $\frac{\partial r_{11}}{\partial c_3} = -\frac{2c_3}{K} - \frac{2c_3(1+c_1^2-c_2^2-c_3^2)}{K^2}$
- $\frac{\partial r_{21}}{\partial c_1} = \frac{2c_2}{K} - \frac{4c_1(c_1c_2+c_3)}{K^2}$
- $\frac{\partial r_{21}}{\partial c_2} = \frac{2c_1}{K} - \frac{4c_2(c_1c_2+c_3)}{K^2}$
- $\frac{\partial r_{21}}{\partial c_3} = \frac{2}{K} - \frac{4c_3(c_1c_2+c_3)}{K^2}$
- $\frac{\partial r_{31}}{\partial c_1} = \frac{2c_3}{K} - \frac{4c_1(c_1c_3-c_2)}{K^2}$
- $\frac{\partial r_{31}}{\partial c_2} = -\frac{2}{K} + \frac{4c_2(c_1c_3-c_2)}{K^2}$
- $\frac{\partial r_{31}}{\partial c_3} = \frac{2c_1}{K} - \frac{4c_3(c_1c_3-c_2)}{K^2}$

the entries of the matrices $\mathbf{A}_2$ are respectively,

- $\frac{\partial r_{12}}{\partial c_1} = \frac{2c_2}{K} - \frac{4c_1(c_1c_2-c_3)}{K^2}$
- $\frac{\partial r_{12}}{\partial c_2} = \frac{2c_1}{K} - \frac{4c_2(c_1c_2-c_3)}{K^2}$

- $\frac{\partial r_{12}}{\partial c_3} = \frac{-2}{K} - \frac{4c_3(c_1c_2-c_3)}{K^2}$
- $\frac{\partial r_{22}}{\partial c_1} = \frac{-2c_1}{K} - \frac{2c_1(1-c_1^2+c_2^2-c_3^2)}{K^2}$
- $\frac{\partial r_{22}}{\partial c_2} = \frac{2c_2}{K} - \frac{2c_2(1-c_1^2+c_2^2-c_3^2)}{K^2}$
- $\frac{\partial r_{22}}{\partial c_3} = \frac{-2c_3}{K} - \frac{2c_3(1-c_1^2+c_2^2-c_3^2)}{K^2}$
- $\frac{\partial r_{32}}{\partial c_1} = \frac{2}{K} - \frac{4c_1(c_1+c_2c_3)}{K^2}$
- $\frac{\partial r_{32}}{\partial c_2} = \frac{2c_3}{K} - \frac{4c_2(c_1+c_2c_3)}{K^2}$
- $\frac{\partial r_{32}}{\partial c_3} = \frac{2c_2}{K} - \frac{4c_3(c_1+c_2c_3)}{K^2}$

the entries of the matrices $\mathbf{A}_3$ are respectively,

- $\frac{\partial r_{13}}{\partial c_1} = \frac{2c_3}{K} - \frac{4c_1(c_2+c_1c_3)}{K^2}$
- $\frac{\partial r_{13}}{\partial c_2} = \frac{2}{K} - \frac{4c_2(c_2+c_1c_3)}{K^2}$
- $\frac{\partial r_{13}}{\partial c_3} = \frac{2c_1}{K} - \frac{4c_3(c_2+c_1c_3)}{K^2}$
- $\frac{\partial r_{23}}{\partial c_1} = \frac{-2}{K} - \frac{4c_1(c_2c_3-c_1)}{K^2}$
- $\frac{\partial r_{23}}{\partial c_2} = \frac{2c_3}{K} - \frac{4c_2(c_2c_3-c_1)}{K^2}$
- $\frac{\partial r_{23}}{\partial c_3} = \frac{2c_2}{K} - \frac{4c_3(c_2c_3-c_1)}{K^2}$
- $\frac{\partial r_{33}}{\partial c_1} = \frac{-2c_1}{K} - \frac{2c_1(1-c_1^2-c_2^2+c_3^2)}{K^2}$
- $\frac{\partial r_{33}}{\partial c_2} = \frac{-2c_2}{K} - \frac{2c_2(1-c_1^2-c_2^2+c_3^2)}{K^2}$
- $\frac{\partial r_{33}}{\partial c_3} = \frac{2c_3}{K} - \frac{2c_3(1-c_1^2-c_2^2+c_3^2)}{K^2}$

A.1.2 Derivation on Robust Weight Function Corresponding to the Tukey-Lambda Distribution

When the shape parameter $\lambda = 0$, the probability density function (pdf) of Tukey-Lamba distribution has the closed form as

$$P(x;\mu,k) = \frac{1}{k(e^{\frac{x-\mu}{2k}} + e^{-\frac{x-\mu}{2k}})^2}, \tag{A.6}$$

which is identical to the Logistic distribution. We assume $\mu = 0$ and thus the robust weight function is derived by

$$\begin{aligned}\omega(x) &= -\frac{1}{2x}\frac{\partial \log P(x;k)}{\partial x} \\ &= \frac{1}{2kx}\frac{e^{\frac{x}{k}}-1}{e^{\frac{x}{k}}+1} \\ &= \begin{cases} \frac{1}{2k^2(e^{\frac{x}{k}}+1)}, \text{if} |x| \leq \epsilon \\ \frac{e^{\frac{x}{k}}-1}{2kx(e^{\frac{x}{k}}+1)}, \text{if} |x| > \epsilon \end{cases},\end{aligned} \tag{A.7}$$

where ϵ is a small positive number.

A.2 Derivations in Regards to 3D Reconstruction Using a Stereo Event Camera

A.2.1 Calculation of the Derivatives

The objective function requires to warp every event's location $\mathbf{x}$ in the reference view to each pair of involved stereo observation $\mathbf{x}_1$ and $\mathbf{x}_2$. First, the 3D point $\mathbf{p}$ inducing the event $\mathbf{x}$ is recovered by performing a back-projection, given the inverse depth ρ:

$$\dot{\mathbf{p}} = \frac{1}{\rho}\begin{pmatrix} \mathtt{P}_1 \\ 0 \quad 0 \quad 0 \quad z \end{pmatrix}^{-1}\begin{pmatrix} u \\ v \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{u-p_{13}}{p_{11}\rho} \\ \frac{v-p_{23}}{p_{22}\rho} \\ \frac{1}{\rho} \\ 1 \end{pmatrix},$$

where $\mathtt{P}_1$ is the 3×4 projection matrix of the left event camera. The following calculation is based on the fact that the last column of $\mathtt{P}_1$ is $\mathbf{0}_{3\times 1}$. Transforming $\mathbf{p}$ to the left camera coordinate of an observation out of $\mathcal{S}_{\text{RV}}$ gives,

$$\mathbf{p}_1 = \mathtt{R}\mathbf{p} + \mathbf{t}. \tag{A.8}$$

The warping results are obtained by

$$\dot{\mathbf{x}}_1 = \mathtt{P}_1\mathbf{p}_1, \tag{A.9}$$

$$\dot{\mathbf{x}}_2 = \mathtt{P}_2\mathbf{p}_1. \tag{A.10}$$

Taking the left event camera, for example,

$$u_1 = \frac{A + B\,\rho}{C + D\,\rho}, \tag{A.11}$$

where

$$\begin{aligned} A &= (p_{11}r_{11} + p_{13}r_{31})\frac{u - p_{13}}{p_{11}} \\ &\quad + (p_{11}r_{12} + p_{13}r_{32})\frac{v - p_{23}}{p_{22}} + (p_{11}r_{13} + p_{13}r_{33}), \\ B &= p_{11}t_x + p_{13}t_z + p_{14}, \\ C &= \frac{r_{31}(u - p_{13})}{p_{11}} + \frac{r_{32}(v - p_{23})}{p_{22}} + r_{33}, \\ D &= t_z. \end{aligned} \tag{A.12}$$

Similarly,

$$v_1 = \frac{A' + B'\rho}{C' + D'\rho}, \tag{A.13}$$

with

$$\begin{aligned} A' &= (p_{22}r_{21} + p_{23}r_{31})\frac{u - p_{13}}{p_{11}} \\ &\quad + (p_{22}r_{22} + p_{23}r_{32})\frac{v - p_{23}}{p_{22}} + (p_{22}r_{23} + p_{23}r_{33}), \\ B' &= p_{22}t_y + p_{23}t_z + p_{24}, \\ C' &= C \\ D' &= D. \end{aligned} \tag{A.14}$$

Therefore, the derivatives with respect to inverse depth d are:

$$\begin{aligned} \frac{\partial u}{\partial \rho} &= \frac{BC - AD}{(C + D\rho)^2}, \\ \frac{\partial v}{\partial \rho} &= \frac{B'C' - A'D'}{(C' + D'\rho)^2}. \end{aligned} \tag{A.15}$$

A.2.2 Uncertainty Propagation

Following (6.9) and only considering the temporal residual for simplicity, we have

$$\rho^\star = \rho_k - \frac{1}{\gamma}\underbrace{(J_1 r_1 + J_2 r_2 + \cdots + J_s r_s)}_{s \in \mathcal{S}_{\text{RV}}}, \tag{A.16}$$

where

$$\gamma \doteq \sum_{s \in \mathcal{S}_{\text{RV}}} J_s^2. \tag{A.17}$$

Therefore the derivative of $\rho^\star$ with respect to $\mathbf{r}$ is

$$\frac{\partial \rho^\star}{\partial \mathbf{r}} = -\frac{1}{\gamma}(J_1, J_2, \cdots, J_s). \tag{A.18}$$

Substituting (A.18) in (6.10), the overall uncertainty of the inverse depth is, to first order, given by

$$\begin{aligned}\sigma_{\rho^\star}^2 &\approx \frac{1}{\gamma}(J_1, J_2, \cdots, J_s)\begin{pmatrix}\sigma_r^2 & & & \\ & \sigma_r^2 & & \\ & & \ddots & \\ & & & \sigma_r^2\end{pmatrix}\frac{1}{\gamma}\begin{pmatrix}J_1\\ J_2\\ \vdots\\ J_s\end{pmatrix} \\ &= \frac{\sigma_r^2}{\gamma^2}(J_1^2 + J_2^2 + \cdots + J_s^2) \\ &\overset{(A.17)}{=} \frac{\sigma_r^2}{\sum_{s\in\mathcal{S}_{RV}} J_s^2}.\end{aligned} \tag{A.19}$$

Bibliography

AGARWAL, A.; JAWAHAR, C.; AND NARAYANAN, P., 2005. A survey of planar homography estimation techniques. *Centre for Visual Information Technology, Tech. Rep. II-IIT/TR/2005/12*, (2005). (cited on page 7)

ARUN, K. S.; HUANG, T. S.; AND BLOSTEIN, S. D., 1987. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , 5 (1987), 698–700. (cited on pages 31, 32, 33, 34, and 35)

AUDRAS, C.; COMPORT, A.; MEILLAND, M.; AND RIVES, P., 2011. Real-time dense appearance-based slam for RGB-D sensors. In *Australasian Conf. on Robotics and Automation*, vol. 2, 2–2. (cited on page 57)

BENOSMAN, R.; IENG, S.-H.; ROGISTER, P.; AND POSCH, C., 2011. Asynchronous event-based Hebbian epipolar geometry. 22, 11 (2011), 1723–1734. doi:10.1109/TNN.2011.2167239. (cited on pages 90 and 104)

BERTHILSSON, R.; ASTROM, K.; AND HEYDEN, A., 2001. Reconstruction of general curves, using factorization and bundle adjustment. *International Journal of Computer Vision (IJCV)*, 41, 3 (2001), 171–182. (cited on page 59)

BESL, P. J. AND MCKAY, N. D., 1991. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14, 2 (1991), 239—-256. (cited on page 59)

BESL, P. J. AND MCKAY, N. D., 1992. Method for registration of 3-d shapes. In *Robotics-DL tentative*, 586–606. International Society for Optics and Photonics. (cited on pages 4, 8, 27, 29, 41, and 42)

BOYKOV, Y.; VEKSLER, O.; AND ZABIH, R., 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23, 11 (2001), 1222–1239. (cited on page 108)

BRANDLI, C.; BERNER, R.; YANG, M.; LIU, S.-C.; AND DELBRUCK, T., 2014. A 240x180 130dB 3us latency global shutter spatiotemporal vision sensor. 49, 10 (2014), 2333–2341. doi:10.1109/JSSC.2014.2342715. (cited on pages xix, 96, and 97)

BYLOW, E.; STURM, J.; KERL, C.; KAHL, F.; AND CREMERS, D., 2013. Direct camera pose tracking and mapping with signed distance functions. In *Demo Track of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems (RSS)*. Berlin, Germany. (cited on pages 59 and 66)

CAMUNAS-MESA, L. A.; SERRANO-GOTARREDONA, T.; IENG, S. H.; BENOSMAN, R. B.; AND LINARES-BARRANCO, B., 2014. On the use of orientation filters for 3D reconstruction in event-driven stereo vision. 8 (2014), 48. doi:10.3389/fnins.2014.00048. (cited on pages 88, 89, and 92)

CARREIRA-PERPIÑÁN, M. Á., 2015. A review of mean-shift algorithms for clustering. *arXiv preprint arXiv:1503.00687*, (2015). (cited on page 30)

CASHMAN, T. J. AND FITZGIBBON, A. W., 2013. What shape are dolphins? building 3d morphable models from 2d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35, 1 (2013), 232—-244. (cited on page 59)

CAYLEY, A., 1846. "about the algebraic structure of the orthogonal group and the other classical groups in a field of characteristic zero or a prime characteristic". In *Reine Angewandte Mathematik*. (cited on page 62)

CENSI, A. AND SCARAMUZZA, D., 2014. Low-latency event-based visual odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. doi:10.1109/IROS.2016.7758089. (cited on pages 13 and 88)

CHAMPLEBOUX, G.; LAVALLEE, S.; SZELISKI, R.; AND BRUNIE, L., 1992. From accurate range imaging sensor calibration to accurate model-based 3d object localization. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, 83–89. IEEE. (cited on page 64)

CHEN, P. AND SUTER, D., 2009. Rank constraints for homographies over two views: revisiting the rank four constraint. *International journal of computer vision*, 81, 2 (2009), 205–225. (cited on page 25)

CHEN, Y. AND MEDIONI, G., 1992. Object modelling by registration of multiple range images. *Image and vision computing*, 10, 3 (1992), 145–155. (cited on pages 58, 59, and 64)

CHENG, Y.; MAIMONE, M.; AND MATTHIES, L., 2005. Visual odometry on the mars exploration rovers. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 903–910. IEEE. (cited on page 2)

CHOJNACKI, W.; SZPAK, Z. L.; BROOKS, M. J.; AND VAN DEN HENGEL, A., 2015. Enforcing consistency constraints in uncalibrated multiple homography estimation using latent variables. *Machine Vision and Applications*, 26, 2-3 (2015), 401–422. (cited on page 25)

CHUI, H. AND RANGARAJAN, A., 2000a. A feature registration framework using mixture models. In *Mathematical Methods in Biomedical Image Analysis, 2000. Proceedings. IEEE Workshop on*, 190–197. IEEE. (cited on pages 8 and 42)

CHUI, H. AND RANGARAJAN, A., 2000b. A new algorithm for non-rigid point matching. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, 44–51. IEEE. (cited on page 41)

COMPORT, A. I.; MALIS, E.; AND RIVES, P., 2007. Accurate quadrifocal tracking for robust 3D visual odometry. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 40–45. IEEE. (cited on page 2)

CONRADT, J.; COOK, M.; BERNER, R.; LICHTSTEINER, P.; DOUGLAS, R. J.; AND DELBRUCK, T., 2009. A pencil balancing robot using a pair of AER dynamic vision sensors. 781–784. doi:10.1109/ISCAS.2009.5117867. (cited on pages 13 and 88)

CORKE, P.; DETWEILER, C.; DUNBABIN, M.; HAMILTON, M.; RUS, D.; AND VASILESCU, I., 2007. Experiments with underwater robot localization and tracking. In *Robotics and Automation, 2007 IEEE International Conference on*, 4556–4561. IEEE. (cited on page 1)

COUGHLAN, J. M. AND YUILLE, A. L., 1999. Manhattan world: Compass direction from a single image by bayesian inference. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 941–947. IEEE. (cited on pages 9, 10, 27, and 43)

COURBON, J.; MEZOUAR, Y.; GUENARD, N.; AND MARTINET, P., 2009. Visual navigation of a quadrotor aerial vehicle. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 5315–5320. IEEE. (cited on page 1)

DA COSTA BOTELHO, S. S.; DREWS, P.; OLIVEIRA, G. L.; AND DA SILVA FIGUEIREDO, M., 2009. Visual odometry and mapping for underwater autonomous vehicles. In *Robotics Symposium (LARS), 2009 6th Latin American*, 1–6. IEEE. (cited on page 1)

DAVISON, A. J., 2003. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE Internation Conference on*, 1403–1410. IEEE. (cited on page 2)

DAVISON, A. J.; REID, I. D.; MOLTON, N. D.; AND OLIVIER, S., 2007. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29, 6 (2007), 1052–1067. (cited on page 2)

DELBRUCK, T. AND LANG, M., 2013. Robotic goalie with 3ms reaction time at 4% CPU load using event-based dynamic vision sensor. 7, 223 (2013). doi:10.3389/fnins.2013.00223. (cited on pages 13 and 88)

DUBROFSKY, E., 2009. *Homography estimation*. Ph.D. thesis, UNIVERSITY OF BRITISH COLUMBIA (Vancouver. (cited on page 22)

EADE, E. AND DRUMMOND, T., 2009. Edge landmarks in monocular slam. *Image and Vision Computing*, 27, 5 (2009), 588–596. (cited on pages 10 and 59)

ENGEL, J.; KOLTUN, V.; AND CREMERS, D., 2017. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 4 (2017). (cited on page 4)

ENGEL, J.; SCHÖPS, T.; AND CREMERS, D., 2014. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, 834–849. Springer. (cited on pages 11, 43, 57, and 59)

ENGEL, J.; STURM, J.; AND CREMERS, D., 2013. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE international conference on computer vision*, 1449–1456. (cited on pages 4, 11, 43, 57, 59, 73, and 105)

FABBRI, R.; COSTA, L. D. F.; TORELLI, J. C.; AND BRUNO, O. M., 2008. 2D euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys (CSUR)*, 40, 1 (2008), 2. (cited on pages 62 and 64)

FABBRI, R. AND KIMIA, B., 2010. 3D curve sketch: Flexible curve-based stereo reconstruction and calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 59)

FAUGERAS, O. AND MOURRAIN, B., 1995. On the geometry and algebra of the point and line correspondences between n images. In *Proceedings of the International Conference on Computer Vision (ICCV)*. (cited on page 58)

FELDMAR, J.; BETTING, F.; AND AYACHE, N., 1995. 3D-2D projective registration of free-form curves and surfaces. In *Proceedings of the International Conference on Computer Vision (ICCV)*. (cited on page 58)

FELZENSZWALB, P. AND HUTTENLOCHER, D., 2004. Distance transforms of sampled functions. Technical report, Cornell University. (cited on page 11)

FELZENSZWALB, P. AND HUTTENLOCHER, D., 2012. Distance transforms of sampled functions. *Theory of Computing*, 8, 19 (2012), 415–428. (cited on page 64)

FISCHLER, M. A. AND BOLLES, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 6 (1981), 381–395. (cited on page 23)

FITZGIBBON, A. W., 2003. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21, 13 (2003), 1145–1153. (cited on pages 8 and 43)

FORSTER, C.; LYNEN, S.; KNEIP, L.; AND SCARAMUZZA, D., 2013. Collaborative monocular slam with multiple micro aerial vehicles. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 3962–3970. IEEE. (cited on page 1)

FORSTER, C.; PIZZOLI, M.; AND SCARAMUZZA, D., 2014. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE Internation Conference on*, 15–22. IEEE. (cited on page 4)

FRISKEN, P. AND ROCKWOOD, J., 2000. Adaptively sampled distance fields: A general representation of shape for computer graphics. In *ACM Transactions on Graphics (SIGGRAPH)*. (cited on pages 59 and 69)

FUKUNAGA, K. AND HOSTETLER, L. D., 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21, 1 (1975), 32–40. (cited on page 43)

FURGALE, P.; REHDER, J.; AND SIEGWART, R., 2013. Unified temporal and spatial calibration for multi-sensor systems. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. (cited on page 96)

GALLEGO, G.; LUND, J. E. A.; MUEGGLER, E.; REBECQ, H.; DELBRUCK, T.; AND SCARAMUZZA, D., 2017. Event-based, 6-DOF camera tracking from photometric depth maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, (2017). (cited on page 13)

GLOVER, J.; BRADSKI, G.; AND RUSU, R. B., 2012. Monte carlo pose estimation with quaternion kernels and the bingham distribution. In *Robotics: science and systems*, vol. 7, 97. (cited on page 27)

GUERRERO, J. AND SAGUES, C., 2001. From lines to homographies between uncalibrated images. In *IX Symposium on Pattern Recognition and Image Analysis, VO4*, 233–240. (cited on page 22)

GUERRERO, J. J. AND SAGÜÉS, C., 2003. Robust line matching and estimate of homographies simultaneously. In *Pattern Recognition and Image Analysis*, 297–307. Springer. (cited on page 22)

HANDA, A.; WHELAN, T.; MCDONALD, J.; AND DAVISON, A., 2014. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*. Hong Kong, China. (cited on pages 73 and 78)

HARRIS, C. AND STEPHENS, M., 1988. A combined corner and edge detector. In *Alvey vision conference*, vol. 15, 10–5244. Citeseer. (cited on page 23)

HARTLEY, R., 1997. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19, 6 (1997), 580–593. (cited on pages 3, 6, 19, and 20)

HARTLEY, R. AND ZISSERMAN, A., 2003. *Multiple view geometry in computer vision*. Cambridge university press. (cited on pages 7 and 88)

HEBB, D. O., 2005. *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press. ISBN 978-0805843002. (cited on page 104)

HIRSCHMULLER, H., 2008. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30, 2 (Feb. 2008), 328–341. doi:10.1109/tpami.2007.1166. (cited on pages 97, 98, and 108)

HOLZ, D.; HOLZER, S.; RUSU, R. B.; AND BEHNKE, S., 2012. Real-time plane segmentation using rgb-d cameras. In *RoboCup 2011: Robot Soccer World Cup XV*, 306–317. Springer. (cited on page 29)

HOSNI, A.; RHEMANN, C.; BLEYER, M.; ROTHER, C.; AND GELAUTZ, M., 2013. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35, 2 (Feb. 2013), 504–511. doi:10.1109/tpami.2012.156. (cited on pages 97 and 98)

HOWARD, A., 2008. Real-time stereo visual odometry for autonomous ground vehicles. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3946–3952. IEEE. (cited on page 2)

JIAN, B. AND VEMURI, B. C., 2011. Robust point set registration using gaussian mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33, 8 (2011), 1633–1645. (cited on pages 8, 41, and 42)

KAESS, M.; ZBOINSKI, R.; AND DELLAERT, F., 2004. MCMC-based multi-view reconstruction of piecewise smooth subdivision curves with a variable number of control points. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on page 59)

KAHL, F. AND AUGUST, J., 2003. Multiview reconstruction of space curves. In *Proceedings of the International Conference on Computer Vision (ICCV)*. (cited on page 59)

KAHL, F. AND HEYDEN, A., 1998. Using conic correspondences in two images to estimate the epipolar geometry. In *Proceedings of the International Conference on Computer Vision (ICCV)*. (cited on page 58)

KAMINSKI, J. Y. AND SHASHUA, A., 2004. Multiple view geometry of general algebraic curves. *International Journal of Computer Vision (IJCV)*, 56, 3 (2004), 195—219. (cited on page 58)

KERL, C.; STURM, J.; AND CREMERS, D., 2013a. Robust odometry estimation for rgb-d cameras. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Karlsruhe, Germany. (cited on page 4)

KERL, C.; STURM, J.; AND CREMERS, D., 2013b. Robust odometry estimation for rgb-d cameras. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 3748–3754. IEEE. (cited on pages 29, 38, 43, 53, 57, 70, and 73)

KIM, H.; LEUTENEGGER, S.; AND DAVISON, A. J., 2016. Real-time 3D reconstruction and 6-DoF tracking with an event camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 349–364. doi:10.1007/978-3-319-46466-4_21. (cited on pages 13 and 88)

KIRYATI, R. K. N. AND BRUCKSTEIN, A. M., 1996. Distance maps and weighted distance transforms. *Journal of Mathematical Imaging and Vision, Special Issue on Topology and Geometry in Computer Vision*, 6 (1996), 223–233. (cited on pages 58 and 59)

KLEIN, G. AND MURRAY, D., 2007. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, 225–234. IEEE. (cited on pages 3, 57, and 73)

KLEIN, G. AND MURRAY, D., 2008. Improving the agility of keyframe-based slam. In *European Conference on Computer Vision*, 802–815. Springer. (cited on page 10)

KNEIP, L.; ZHOU, Y.; AND LI, H., 2015. Sdicp: Semi-dense tracking based on iterative closest points. In *Proceedings of the British Machine Vision Conference (BMVC)*, 100.1–100.12. BMVA Press. doi:10.5244/C.29.100. https://dx.doi.org/10.5244/C.29.100. (cited on pages 11, 15, 43, 59, 62, and 63)

KOGLER, J.; HUMENBERGER, M.; AND SULZBACHNER, C., 2011. Event-based stereo matching approaches for frameless address event stereo data. 674–685. doi:10.1007/978-3-642-24028-7_62. (cited on pages 88, 89, and 92)

KOŠECKÁ, J. AND ZHANG, W., 2002. Video compass. In *Computer Vision—ECCV 2002*, 476–490. Springer. (cited on pages 9, 10, and 43)

KUENG, B.; MUEGGLER, E.; GALLEGO, G.; AND SCARAMUZZA, D., 2016. Low-latency visual odometry using event-based feature tracks. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 16–23. Daejeon, Korea. doi:10.1109/IROS.2016.7758089. (cited on pages 13 and 88)

KUSE, M. P. AND SHEN, S., 2016. Robust camera motion estimation using direct edge alignment and sub-gradient method. In *IEEE International Conference on Robotics and Automation (ICRA-2016), Stockholm, Sweden*. (cited on pages 11, 12, 59, 62, 63, 74, and 75)

LACROIX, S.; MALLET, A.; CHATILA, R.; AND GALLO, L., 1999. Rover self localization in planetary-like environments. In *Artificial Intelligence, Robotics and Automation in Space*, vol. 440, 433. (cited on pages 1 and 2)

LAGORCE, X.; MEYER, C.; IENG, S.-H.; FILLIAT, D.; AND BENOSMAN, R., 2015. Asynchronous event-based multikernel algorithm for high-speed visual features tracking. 26, 8 (Aug. 2015), 1710–1720. doi:10.1109/TNNLS.2014.2352401. (cited on pages 13 and 88)

LAGORCE, X.; ORCHARD, G.; GALLUPI, F.; SHI, B. E.; AND BENOSMAN, R., 2016. HOTS: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, PP, 99 (2016), 1–1. doi:10.1109/TPAMI.2016.2574707. (cited on page 90)

LANGELAAN, J. AND ROCK, S., 2005. Towards autonomous uav flight in forests. In *Proc. of AIAA Guidance, Navigation and Control Conference*. (cited on page 1)

LEPETIT, V.; MORENO-NOGUER, F.; AND FUA, P., 2009. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81, 2 (2009), 155. (cited on page 105)

LICHTSTEINER, P.; POSCH, C.; AND DELBRUCK, T., 2008. A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. 43, 2 (2008), 566–576. doi:10.1109/JSSC.2007.914337. (cited on pages 13 and 87)

LONGUET-HIGGINS, H. C., 1987. A computer algorithm for reconstructing a scene from two projections. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds*, (1987), 61–62. (cited on page 6)

LOWE, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 2 (2004), 91–110. (cited on page 5)

LU, Y. AND SONG, D., 2015. Robustness to lighting variations: An RGB-D indoor visual odometry using line segments. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 688–694. IEEE. (cited on pages 10 and 38)

LUONG, Q. AND FAUGERAS, O., 1993. Determining the fundamental matrix with planes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 489–494. (cited on pages 6 and 18)

LUONG, Q.-T. AND FAUGERAS, O. D., 1996. The fundamental matrix: Theory, algorithms, and stability analysis. *International journal of computer vision*, 17, 1 (1996), 43–75. (cited on page 6)

LYNEN, S.; SATTLER, T.; BOSSE, M.; HESCH, J.; POLLEFEYS, M.; AND SIEGWART, R., 2015. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Proceedings of Robotics: Science and Systems*. Rome, Italy. doi:10.15607/RSS.2015.XI.037. (cited on page 106)

MARR, D. AND POGGIO, T., 1976. Cooperative computation of stereo disparity. *Science*, 194, 4262 (1976), 283–287. (cited on page 88)

MATTHIES, L. AND SHAFER, S., 1987. Error modeling in stereo navigation. *IEEE Journal on Robotics and Automation*, 3, 3 (1987), 239–248. (cited on page 2)

MATTHIES, L. H., 1989. Dynamic stereo vision. (1989). (cited on page 2)

MILELLA, A. AND SIEGWART, R., 2006. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, 21–21. IEEE. (cited on page 2)

MONTEMERLO, M.; THRUN, S.; KOLLER, D.; WEGBREIT, B.; ET AL., 2002. Fastslam: A factored solution to the simultaneous localization and mapping problem. *AAAI/IAAI*, 593–598 (2002). (cited on page 2)

MORAVEC, H. P., 1980. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document. (cited on pages 1 and 2)

MUEGGLER, E.; HUBER, B.; AND SCARAMUZZA, D., 2014. Event-based, 6-DOF pose tracking for high-speed maneuvers. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2761–2768. doi:10.1109/IROS.2014.6942940. (cited on pages 13 and 88)

MUEGGLER, E.; REBECQ, H.; GALLEGO, G.; DELBRUCK, T.; AND SCARAMUZZA, D., 2017. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *International Journal of Robotics Research (IJRR)*, 36 (2017), 142–149. doi:10.1177/0278364917691115. (cited on pages 96, 97, and 98)

MUR-ARTAL, R.; MONTIEL, J.; AND TARDÓS, J. D., 2015. ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31, 5 (2015), 1147–1163. (cited on pages 3 and 57)

MUR-ARTAL, R. AND TARDOS, J. D., 2015. Probabilistic semi-dense mapping from highly accurate feature-based monocular SLAM. In *Proceedings of Robotics: Science and Systems (RSS)*. (cited on page 95)

MUR-ARTAL, R. AND TARDÓS, J. D., 2017. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33, 5 (2017), 1255–1262. doi:10.1109/TRO.2017.2705103. (cited on pages 75 and 81)

NEWCOMBE, R. A.; IZADI, S.; HILLIGES, O.; MOLYNEAUX, D.; KIM, D.; DAVISON, A.; KOHI, P.; SHOTTON, J.; HODGES, S.; AND FITZGIBBON, A., 2011a. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, 127–136. IEEE. (cited on pages 4, 57, 59, and 66)

NEWCOMBE, R. A.; LOVEGROVE, S. J.; AND DAVISON, A. J., 2011b. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, 2320–2327. IEEE. (cited on pages 3, 11, 43, and 89)

NILSSON, D.-E., 1996. Eye ancestry: old genes for new eyes. *Current Biology*, 6, 1 (1996), 39–42. (cited on page 1)

NISTÉR, D., 2004. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26, 6 (2004), 756–777. (cited on page 3)

NISTÉR, D.; NARODITSKY, O.; AND BERGEN, J., 2004. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 652–659. Washington, DC, USA. (cited on pages 1, 2, and 3)

NURUTDINOVA, I. AND FITZGIBBON, A., 2015. Towards pointless structure from motion: 3d reconstruction and camera parameters from general 3d curves. In *Proceedings of the IEEE International Conference on Computer Vision*, 2363–2371. (cited on pages 11, 12, 58, 59, 63, 66, and 73)

OLSON, C. F.; MATTHIES, L. H.; SCHOPPERS, H.; AND MAIMONE, M. W., 2000. Robust stereo ego-motion for long distance navigation. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, 453–458. IEEE. (cited on page 2)

OLSON, C. F.; MATTHIES, L. H.; SCHOPPERS, M.; AND MAIMONE, M. W., 2003. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43, 4 (2003), 215–229. (cited on page 2)

PARKER, A., 2016. *In the blink of an eye: how vision kick-started the big bang of evolution*. Natural History Museum. (cited on page 1)

PIATKOWSKA, E.; BELBACHIR, A. N.; AND GELAUTZ, M., 2013. Asynchronous stereo vision for event-driven dynamic stereo sensor using an adaptive cooperative approach. In *iccvw*, 45–50. (cited on page 88)

POMERLEAU, F.; COLAS, F.; SIEGWART, R.; AND MAGNENAT, S., 2013. Comparing icp variants on real-world data sets. *Autonomous Robots*, 34, 3 (2013), 133–148. (cited on pages 4, 29, 38, 41, 43, 53, 57, and 59)

POMERLEAU, F.; MAGNENAT, S.; COLAS, F.; LIU, M.; AND SIEGWART, R., 2011. Tracking a depth camera: Parameter exploration for fast icp. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3824–3829. IEEE. (cited on pages 4, 29, 38, 41, 43, 53, 57, and 59)

PORRILL, J. AND POLLARD, S., 1991. Curve matching and stereo calibration. *Image and Vision Computing (IVC)*, 9, 1 (1991), 45–50. (cited on page 58)

PRITCHETT, P. AND ZISSERMAN, A., 1998. Matching and reconstruction from widely separated views. In *3D Structure from Multiple Images of Large-Scale Environments*, 78–92. Springer. (cited on page 7)

PUPILLI, M. AND CALWAY, A., 2005. Real-time camera tracking using a particle filter. In *BMVC*. (cited on page 2)

PUPILLI, M. AND CALWAY, A., 2006. Real-time visual slam with resilience to erratic motion. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, 1244–1249. IEEE. (cited on page 2)

REBECQ, H.; GALLEGO, G.; MUEGGLER, E.; AND SCARAMUZZA, D., 2017a. EMVS: Event-based multi-view stereo. *International Journal of Computer Vision (IJCV)*, (2017). Under review. (cited on pages xix, 88, 90, 98, and 100)

REBECQ, H.; GALLEGO, G.; MUEGGLER, E.; AND SCARAMUZZA, D., 2017b. Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time. *International Journal of Computer Vision*, (2017), 1–21. (cited on pages xv and 14)

REBECQ, H.; GALLEGO, G.; AND SCARAMUZZA, D., 2016. EMVS: Event-based multi-view stereo. In *Proceedings of the British Machine Vision Conference (BMVC)*. (cited on pages 88 and 89)

REBECQ, H.; HORSTSCHÄFER, T.; GALLEGO, G.; AND SCARAMUZZA, D., 2017c. EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time. 2 (2017), 593–600. doi:10.1109/LRA.2016.2645143. (cited on pages 13 and 88)

ROBERTS, L., 1965. *Machine perception of 3-d solids*. Ph.D. thesis, MIT. (cited on page 57)

ROGISTER, P.; BENOSMAN, R.; IENG, S.-H.; LICHTSTEINER, P.; AND DELBRUCK, T., 2012. Asynchronous event-based binocular stereo matching. 23, 2 (2012), 347–353. doi:10.1109/TNNLS.2011.2180025. (cited on pages 88, 89, and 92)

ROSINOL VIDAL, A.; REBECQ, H.; HORSTSCHAEFER, T.; AND SCARAMUZZA, D., 2018. Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios. 3, 2 (April 2018), 994–1001. doi:10.1109/LRA.2018.2793357. (cited on page 13)

RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; AND BRADSKI, G., 2011. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, 2564–2571. IEEE. (cited on page 3)

SANCHEZ, J.; DENIS, F.; CHECCHIN, P.; DUPONT, F.; AND TRASSOUDAINE, L., 2017. Global registration of 3d lidar point clouds based on scene features: Application to structured environments. *Remote Sensing*, 9, 10 (2017), 1014. (cited on pages xv and 9)

SCARAMUZZA, D. AND SIEGWART, R., 2008. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE transactions on robotics*, 24, 5 (2008), 1015–1026. (cited on page 4)

SCHOEPS, T.; ENGEL, J.; AND CREMERS, D., 2014. Semi-dense visual odometry for ar on a smartphone. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE Internation Symposium on*, 145–150. IEEE. (cited on pages 11 and 43)

SCHRAML, S.; BELBACHIR, A. N.; AND BISCHOF, H., 2015. Event-driven stereo matching for real-time 3D panoramic vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 466–474. doi:10.1109/CVPR.2015.7298644. (cited on pages 88 and 89)

SCHRAML, S.; BELBACHIR, A. N.; AND BISCHOF, H., 2016. An event-driven stereo system for real-time 3-D 360 panoramic vision. 63, 1 (Jan. 2016), 418–428. doi:10.1109/tie.2015.2477265. (cited on pages 88 and 89)

SCHWARZ, M.; SCHULZ, H.; AND BEHNKE, S., 2015. Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 1329–1335. IEEE. (cited on page 27)

SHASHUA, A. AND AVIDAN, S., 1996. The rank 4 constraint in multiple view geometry. In *Computer Vision—ECCV'96*, 196–206. Springer. (cited on page 25)

SILBERMAN, N.; HOIEM, D.; KOHLI, P.; AND FERGUS, R., 2012. Indoor segmentation and support inference from RGB-D images. In *Computer Vision–ECCV 2012*, 746–760. Springer. (cited on pages 9 and 43)

SINCLAIR, D.; CHRISTENSEN, H.; AND ROTHWELL, C., 1995. Using the relation between a plane projectivity and the fundamental matrix. In *Proc. SCIA*, 181–188. (cited on page 7)

SON, B.; SUH, Y.; KIM, S.; JUNG, H.; KIM, J.-S.; SHIN, C.; PARK, K.; LEE, K.; PARK, J.; WOO, J.; ET AL., 2017. 4.1 a 640× 480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation. In *Solid-State Circuits Conference (ISSCC), 2017 IEEE International*, 66–67. IEEE. (cited on page 108)

SRINIVASAN, M.; ZHANG, S.; AND BIDWELL, N., 1997. Visually mediated odometry in honeybees. *Journal of Experimental Biology*, 200, 19 (1997), 2513–2522. (cited on page 2)

STEINBRÜCKER, F.; KERL, C.; STURM, J.; AND CREMERS, D., 2013. Large-scale multi-resolution surface reconstruction from RGB-D sequences. In *Proceedings of the International Conference on Computer Vision (ICCV)*. (cited on pages 59 and 66)

STEINBRÜCKER, F.; STURM, J.; AND CREMERS, D., 2011. Real-time visual odometry from dense rgb-d images. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 719–722. IEEE. (cited on page 57)

STEWART, C. V., 1999. Robust parameter estimation in computer vision. *SIAM review*, 41, 3 (1999), 513–537. (cited on page 6)

STRAUB, J.; BHANDARI, N.; LEONARD, J. J.; AND FISHER III, J. W., 2015a. Real-time manhattan world rotation estimation in 3d. In *IROS*. http://www.jstraub.de/download/straub2015rtmf.pdf. (cited on pages 9, 10, and 43)

STRAUB, J.; BHANDARI, N.; LEONARD, J. J.; AND FISHER III, J. W., 2015b. Real-time manhattan world rotation estimation in 3d. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 1913–1920. IEEE. (cited on pages 27 and 29)

STRAUB, J.; ROSMAN, G.; FREIFELD, O.; LEONARD, J. J.; AND FISHER, J. W., 2014. A mixture of manhattan frames: Beyond the manhattan world. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 3770–3777. IEEE. (cited on pages 9, 27, 29, 43, and 45)

STÜCKLER, J.; STEFFENS, R.; HOLZ, D.; AND BEHNKE, S., 2011. Real-time 3d perception and efficient grasp planning for everyday manipulation tasks. In *ECMR*, 177–182. (cited on page 27)

STURM, J.; ENGELHARD, N.; ENDRES, F.; BURGARD, W.; AND CREMERS, D., 2012. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. (cited on pages 38, 53, 73, and 74)

SZELISKI, R. AND TORR, P. H., 1998. Geometrically constrained structure from motion: Points on planes. In *3D Structure from Multiple Images of Large-Scale Environments*, 171–186. Springer. (cited on pages 5, 7, and 17)

SZPAK, Z. L.; CHOJNACKI, W.; ERIKSSON, A.; AND VAN DEN HENGEL, A., 2014. Sampson distance based joint estimation of multiple homographies with uncalibrated cameras. *Computer Vision and Image Understanding*, 125 (2014), 200–213. (cited on pages xv, 7, 23, and 25)

TAGUCHI, Y.; JIAN, Y.-D.; RAMALINGAM, S.; AND FENG, C., 2013. Point-plane slam for hand-held 3d sensors. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 5182–5189. IEEE. (cited on pages 9 and 43)

TAN, W.; LIU, H.; DONG, Z.; ZHANG, G.; AND BAO, H., 2013. Robust monocular slam in dynamic environments. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, 209–218. IEEE. (cited on page 106)

TANSKANEN, P.; KOLEV, K.; MEIER, L.; CAMPOSECO, F.; SAURER, O.; AND POLLEFEYS, M., 2013. Live metric 3D reconstruction on mobile phones. In *Proceedings of the IEEE International Conference on Computer Vision*, 65–72. (cited on page 73)

TENEY, D. AND PIATER, J., 2012. Sampling-based multiview reconstruction without correspondences for 3D edges. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*. (cited on page 59)

TOMIC, T.; SCHMID, K.; LUTZ, P.; DOMEL, A.; KASSECKER, M.; MAIR, E.; GRIXA, I. L.; RUESS, F.; SUPPA, M.; AND BURSCHKA, D., 2012. Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *Robotics & Automation Magazine, IEEE*, 19, 3 (2012), 46–56. (cited on page 1)

TREVOR, A. J.; ROGERS III, J. G.; CHRISTENSEN, H.; ET AL., 2012. Planar surface slam with 3d and 2d sensors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 3041–3048. IEEE. (cited on pages 9 and 43)

TRIGGS, B.; MCLAUCHLAN, P.; HARTLEY, R.; AND FITZGIBBON, A., 1999. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice (ICCV)*, 298–372. Corfu, Greece. (cited on page 3)

TSIN, Y. AND KANADE, T., 2004. A correlation-based approach to robust point set registration. In *Computer Vision-ECCV 2004*, 558–569. Springer. (cited on pages 8 and 42)

TYKKÄLÄ, T.; AUDRAS, C.; AND COMPORT, A. I., 2011. Direct iterative closest point for real-time visual odometry. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2050–2056. IEEE. (cited on pages 57 and 59)

VINCENT, E. AND LAGANIÉRE, R., 2001. Detecting planar homographies in an image pair. In *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, 182–187. IEEE. (cited on page 7)

WEI, X.; PHUNG, S. L.; AND BOUZERDOUM, A., 2014. Object segmentation and classification using 3-d range camera. *Journal of Visual Communication and Image Representation*, 25, 1 (2014), 74–85. (cited on page 27)

WEIKERSDORFER, D.; ADRIAN, D. B.; CREMERS, D.; AND CONRADT, J., 2014. Event-based 3D SLAM with a depth-augmented dynamic vision sensor. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 359–364. doi:10.1109/ICRA.2014.6906882. (cited on pages 13 and 88)

WEINGARTEN, J. AND SIEGWART, R., 2006. 3d slam using planar segments. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 3062–3067. IEEE. (cited on pages 9 and 43)

WHELAN, T.; KAESS, M.; FALLON, M.; JOHANNSSON, H.; LEONARD, J.; AND MCDONALD, J., 2012a. Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*. Sydney, Australia. (cited on page 4)

WHELAN, T.; KAESS, M.; FALLON, M.; JOHANNSSON, H.; LEONARD, J.; AND MCDONALD, J., 2012b. Kintinuous: Spatially extended kinectfusion. In *3rd RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras, (Sydney, Australia)*. (cited on page 57)

WIKIPEDIA.ORG. Iteratively reweighted least squares. https://en.wikipedia.org/wiki/Iteratively_reweighted_least_squares. (cited on page 33)

XIAO, Y. J. AND LI, Y., 2005. Optimized stereo reconstruction of free-form space curves based on a nonuniform rational B-spline model. *Journal of the Optical Society of America*, 22, 9 (2005), 1746–1762. (cited on pages 58 and 59)

YANG, J. AND LI, H., 2015. Dense, accurate optical flow estimation with piecewise parametric model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1019–1027. (cited on page 106)

YANG, J.; LI, H.; CAMPBELL, D.; AND JIA, Y., 2016. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38, 11 (2016), 2241–2254. (cited on page 59)

YANG, J.; LI, H.; AND JIA, Y., 2013. Go-icp: Solving 3D registration efficiently and globally optimally. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 1457–1464. IEEE. (cited on pages 4, 8, 29, and 43)

ZEINIK-MANOR, L. AND IRANI, M., 2002. Multiview constraints on homographies. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24, 2 (2002), 214–223. (cited on page 25)

ZENG, H.; DENG, X.; AND HU, Z., 2008. A new normalized method on line-based homography estimation. *Pattern Recognition Letters*, 29, 9 (2008), 1236–1244. (cited on page 23)

ZHANG, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision (IJCV)*, 13, 12 (1994), 119–152. (cited on page 59)

ZHANG, Z., 1997. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and vision Computing*, 15, 1 (1997), 59–76. (cited on page 70)

ZHANG, Z., 1998. Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*, 27, 2 (1998), 161–195. (cited on page 7)

ZHOU, Y.; GALLEGO, G.; REBECQ, H.; KNEIP, L.; LI, H.; AND SCARAMUZZA, D., 2018a. Semi-dense 3d reconstruction with a stereo event camera. In *European Conference on Computer Vision*, vol. 2. Springer, Cham. (cited on page 15)

ZHOU, Y.; KNEIP, L.; AND LI, H., 2015. A revisit of methods for determining the fundamental matrix with planes. In *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*, 1–7. IEEE. (cited on page 15)

ZHOU, Y.; KNEIP, L.; AND LI, H., 2016a. Real-time rotation estimation for dense depth sensors in piece-wise planar environments. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 2271–2278. IEEE. (cited on page 15)

ZHOU, Y.; KNEIP, L.; AND LI, H., 2017. Semi-dense visual odometry for RGB-D cameras using approximate nearest neighbour fields. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 6261–6268. IEEE. (cited on pages 11 and 15)

ZHOU, Y.; KNEIP, L.; RODRIGUEZ, C.; AND LI, H., 2016b. Divide and conquer: Efficient density-based tracking of 3d sensors in manhattan worlds. In *Asian Conference on Computer Vision (ACCV)*, 3–19. Springer. (cited on page 15)

ZHOU, Y.; LI, H.; AND KNEIP, L., 2018b. Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment. *IEEE Transactions on Robotics*, (2018), 1–16. doi:10.1109/TRO.2018.2875382. (cited on page 15)

ZHU, A. Z.; ATANASOV, N.; AND DANIILIDIS, K., 2017. Event-based feature tracking with probabilistic data association. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. (cited on pages 13 and 88)

ZHU, A. Z.; THAKUR, D.; OZASLAN, T.; PFROMMER, B.; KUMAR, V.; AND DANIILIDIS, K., 2018. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. 3, 3 (Jul. 2018), 2032–2039. doi:10.1109/lra.2018.2800793. (cited on pages 92, 96, 97, and 98)