Direct Pose Estimation and Refinement

Hatem Alismail
halismai@cs.cmu.edu

CMU-RI-TR-16-50

*Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Robotics.*

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

August, 2016

Thesis committee:
Brett Browning, *co-chair*
Simon Lucey, *co-chair*
Michael Kaess
Martial Hebert
Ian D. Reid, The University of Adelaide

Abstract

We study a fundamental question in pose estimation from vision-only video data: should the pose of a camera be determined from fixed and known correspondences? Or should correspondences be simultaneously estimated alongside the pose?

Determining pose from fixed correspondences is known as feature-based, where well-established tools from projective geometry are utilized to formulate and solve a plethora of pose estimation problems. Nonetheless, in degraded imaging conditions such as low light and blur, reliably detecting and precisely localizing interest points becomes challenging.

Conversely, estimating correspondences alongside motion is known as the direct approach, where image data are used directly to determine geometric quantities without relying on sparse interest points as an intermediate representation. The approach is in general more precise by virtue of redundancy as many measurements are used to estimate a few degrees-of-freedom. However, direct methods are more sensitive to changes in illumination.

In this work, we combine the best of the feature-based approaches with the precision of direct methods. Namely, we make use of densely and sparsely evaluated local feature descriptors in a direct image alignment framework to address pose estimation in challenging conditions. Applications include tracking planar targets under sudden and drastic changes in illumination as well as visual odometry in poorly-lit subterranean mines.

Motivated by the success of the proposed approach, we introduce a novel formulation for the joint refinement of pose and structure across multiple views akin to feature-based bundle adjustment (BA). In contrast to minimizing the reprojection error using BA, initial estimates are refined such that the photometric consistency of their image projections is maximized without the need for correspondences. The fundamentally different technique is evaluated on a range of datasets and is shown to improve upon the accuracy of the state-of-the-art in vision-based simultaneous localization and mapping (VSLAM).

To Family and in memory of
Gradnma Salma & Uncle Mufeed.

Acknowledgments

None of this work could have been possible without the guidance, support, and encourgment of many people. First, I would like to thank my graduate education advisors: Brett Browning, Bernardine Dias and Simon Lucey. I have worked with Brett and Bernardine since my sophomore year at Carnegie Mellon Qatar. They did not only teach about Robotics, but also introduced me to the joys of research. Without Brett & Bernardine I would not have been here; thank you and I cannot begin to express my gratitude for all of what you have taught me.

I also had the pleasure and great opportunity to work with Simon Lucey for the last two years in my PhD. His enthusiasm, energy, advise, encouragement and support are phenomenal. I have learned a great deal academically and personally from Simon; Thank you Simon, this dissertaion would not have been possible without your help.

Thanks to my thesis committee for their valuable feedback: Michael Kaess, Martial Hebert and Ian Reid; thank you for being accomdating given the ten and a half hours time zone differnce. Thanks also to Reid Simmons and Martial Hebert (again) for making sure that I was on track for the completion of this dissertaion.

At CMU, I had the pleasure of working at the National Robotics Engineering Center (NREC) for a few years where I have learned a great deal from Peter Rander. Thank you Pete. Thanks also to NREC IT for their excellent support and fixing my laptop not only once, but twice. Thanks to Herman Herman and Kelly Mullins for their help with computing equipments towards the end of my PhD.

One of the many strenghts of CMU beyond academia is the wondeful project and institute staff; this dissertaion (and many others) would not have been possible without their tireless work. Thanks to: Suzanne Lyons Muth, Lynnetta Miller, Sanae Minick, Cindy Glick, Rachel Burcin, Jim Ketterer, Vladimir Altman, Suzette Gambone, Jim Martin, and Jim Montgomery.

Thanks also to the wondeful TechBridgeWorld family: Bernardine Dias, Sarah Belousov, Freddie Dias, Bea Dias and Ermine Teves; your positive influence on the world will live on through the many projects and the many students you taught.

Thanks to members of the CI2CV lab as well: Hamed, Chen-Kong, Chen-Hsuan, Rui, Kit and Ash; it was great sharing an office with you. It was also great sharing an office with Peter and David at NREC.

Thanks to my first Robotics teachers: Chuck Thorpe, Justin Carlson and David Duke (the Amigo Bots remain on sale!). Thanks also to Majd Sakr who taught valuable life lessons.

Graduate school is not only nerding. Thanks to: Tessa, Narek, Saurabh, Abha, Ceren, Aparna, Sid, Melanie, Liqi, Harsimrat, Billy, Anu, Anothony and Heather. Thank you Heather for the valuable suggestions that improved my dissertaion. Thanks also to Rose, Jack and the Russians!

None of this work would have been possible with the loving companionship of Ermine. Your care and patience are highly appreciated; I owe you so much. Thank you Ermine.

Last but not least, spending n years of graduate education would not have been possible without the support and encourgment of my family. I am lucky to have you.

I am sure I forgot to acknowledge many people who contributed vastly to the completion of this dissertaion; thank you all.

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

1.1 Problem and Motivation

The motivating problem of this research is:

> *How to perform vision-only pose estimation from challenging data?*

Challenging data, in this work, is characterized by images captured under poor illumination conditions, such as low light. Pose estimation, or tracking, is used in a broad sense, where we demonstrate results on problems including: (i) tracking the eight degrees-of-freedom of a planar target, (ii) estimating the six degrees-of-freedom between two rigidly moving cameras sharing a common field of view (visual odometry), and (iii) vision-based Simultaneous Localization and Mapping (VSLAM), where we refine initial estimates of the viewing parameters of multiple cameras jointly with the scene structure.

Why vision-only? Why challenging data?

Vision-only pose estimation has advanced immensely in the past few years. It is now possible to perform pose estimation tasks with high precision and high fidelity [132, 322, 327]. Applications of robust and accurate pose estimation from vision data are also broad, especially when interactions with the world are required, which range from intelligent homes [68] and virtual reality [194] to autonomous driving [365, 379] and space exploration [251].

It is not unusual for robotic systems to rely solely on vision. After all, cameras are inexpensive, have low power requirements, and — most important of all — provide a rich

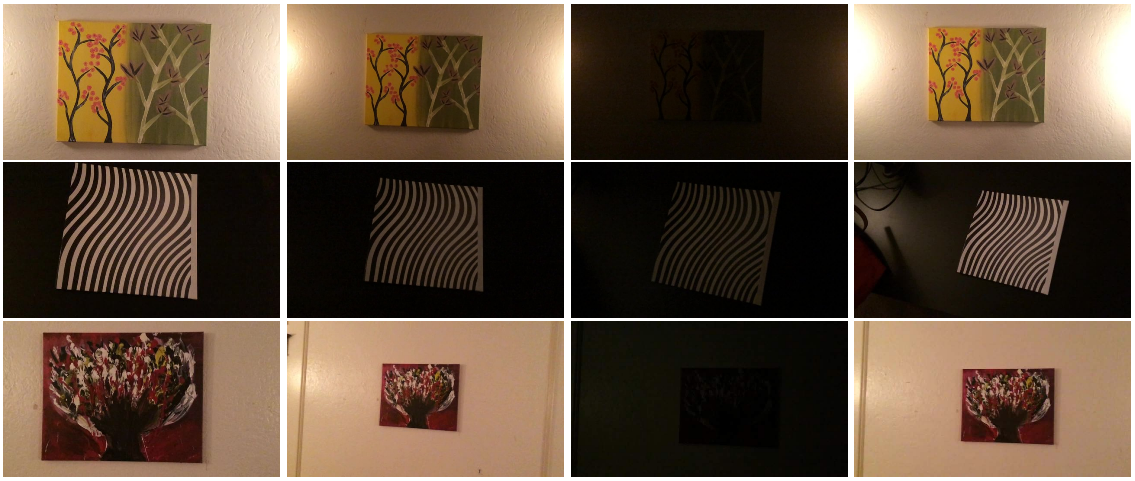


Figure 1.1: *Overview of challenging datasets characterized by sudden and drastic illumination changes as well as ambiguously textured objects. These forms of appearance variations remain challenging for the current state-of-the-art pose estimation algorithms. The datasets are available online at* http://www.cs.cmu.edu/~halismai/bitplanes/.

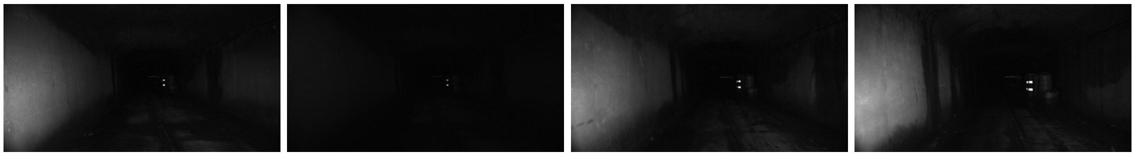


Figure 1.2: *Challenging data in poorly lit underground mines. As mines are pitch dark, the robot must supply its own source of lighting, which is often insufficient to illuminate the scene. In addition, there are persistent changes in appearance due to the camera auto exposure settings, especially when the robot drives in proximity to the mine walls.*

view of the visual world. By pushing the limits of robustness using nothing but imagery, vision-only and vision-assisted systems would benefit alike.

By the same token, enabling vision to work in challenging environments is desired for wide spread and reliable use of vision in robotic application domains [69, 80, 262, 282, 298]. Defining what constitutes "challenging data" is in itself a challenging task. As a matter of course, as research is ever advancing, the frontier of what constitutes challenging advances as well.

We illustrate the term challenging by example. For instance, Fig. 1.2 shows an example of images collected using an underground mining robot. As mines are pitch dark, the robot must supply its own source of illumination. However, due to power and other requirements, the illumination is often insufficient. In addition, there are various nonlinear deformations of image appearance due to the camera auto settings especially when the robot drives in proximity to the mine walls. Consequently, imagery is characterized by a low signal-to-noise ratio causing the state-of-the-art techniques to fail too frequently. This type of data is in fact too challenging at times as images are often over/under saturated — a consequence of the geometry of mines — such that no viable information can be gleaned (as we show in Chapter 6). Nonetheless, the techniques we develop in this work improve robustness measurably.

Another example of challenging imagery includes sudden and drastic illumination changes as shown in Fig. 1.1, where the task is to track paintings across every frame starting from a known template location. The dataset was collected using a hand-held mobile phone during the night, where lights were flicked to create a transition from a sufficiently illuminated environment to near darkness. This sudden and drastic change in appearance remains difficult to address using the current state-of-the-art algorithms. Addressing this type of illumination change is the topic of Chapter 5 and Chapter 6, where we demonstrate not only robust results, but faster than real-time performance on consumer laptops and mobile devices.

The example data in Fig. 1.1 is unconfined to indoor environments. A similar situation is often observed due to automatic camera controls, which require a few seconds to adjust when illumination changes rapidly as illustrated in Fig. 1.3. Similar illumination conditions also arise upon entering and leaving traffic tunnels, for instance.

Finally, an additional form of challenging data is characterized by imaging non-Lambertian surfaces. In particular, specular and shiny objects as shown in Fig. 1.4, which are also challenging and typically treated with special algorithms [175, 283].

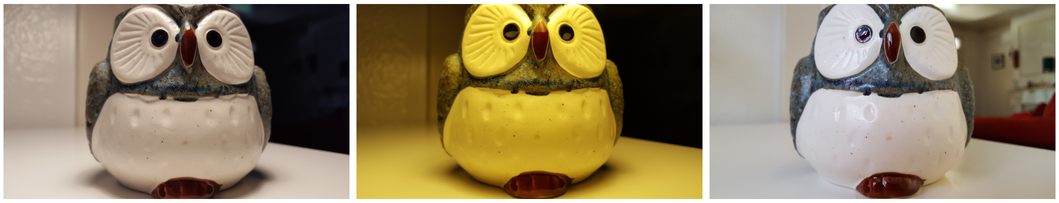


Figure 1.3: Illumination changes due to camera auto gain and auto white balance settings.

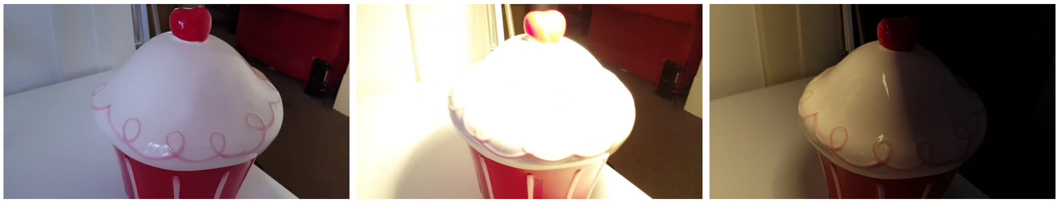


Figure 1.4: Illumination changes due to specular reflections causing over saturation at large parts of the object as well as shadows causing under saturation.

1.2 The Correspondence Problem

At the heart of pose estimation, and computer vision, is the *correspondence problem*. Given a collection of images with overlapping fields of view, the goal of the correspondence problem is to determine which parts of the images correspond to which parts.

In geometric estimation problems, such as the ones considered in this work, merely establishing correspondences is insufficient; correspondences must be established with high accuracy for precise results. To this end, there are two major paradigms for pose estimation distinguished by the way correspondences are established between imagery. One is the feature-based approach, where the image is abstracted into a few keypoint positions [369]. The features, or keypoints, are usually matched using local feature descriptors. Feature matches are then used to estimate the pose using geometry and robust fitting [123, 160]. The other is the direct approach, where image intensities are used directly to determine the desired quantities [181]. In the direct approach, correspondences are automatically determined as a byproduct of the pose estimation procedure.

Both methods have their strengths and weaknesses, which have been previously discussed in the literature [181, 369]. Features — used in the sense of keypoint positions — have proven powerful due to the viewpoint and photometric invariances they afford. In addition, once the image is abstracted away as a collection of keypoint positions, reasoning about the geometry of the 3D world can be simplified with well established tools from projective geometry [119, 160].

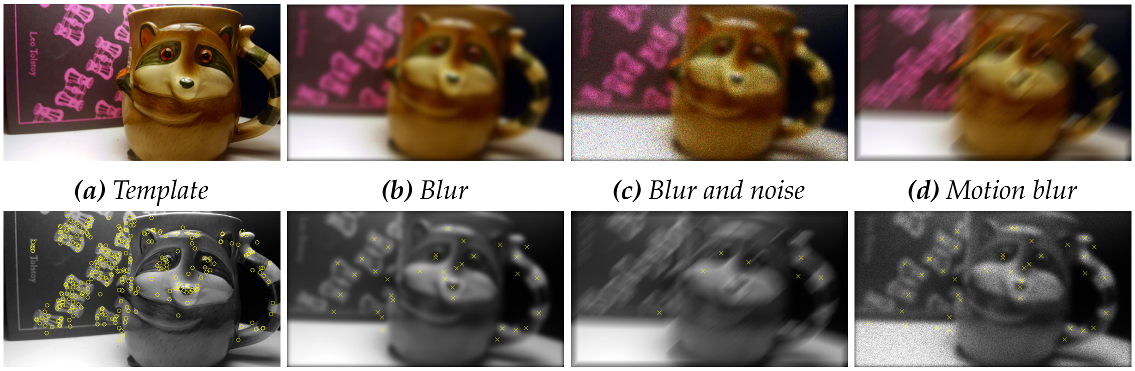
On the other hand, direct methods — also known as image-based, intensity-based, model-based, Lucas-Kanade-like, or photometric — keep the image in the loop. Their traditional uses have been focused on estimating dense correspondences, such as the estimation of the optical flow [171, 352] and stereo [235, 324], but they can be used semi-densely [109], or even sparsely [10, 97]. Recently, however, with the increasing availability of high frame-rate video and the improved computational power, the popularity of direct tracking has been on the rise as a tool for robust camera pose tracking and VSLAM [107, 110, 198].

What makes feature-based methods powerful?

Key to the wide success of feature-based algorithms is the development of *feature descriptors*, which encode image information in a way that provides invariance to a range of geometric and photometric deformations. For instance, the development of the SIFT [234] and the HOG [89] descriptors have made possible challenging correspondence estimation problems, such as Structure-from-Motion (SFM) from unorganized images [5, 339] and large-scale multi-view stereopsis [132, 366].

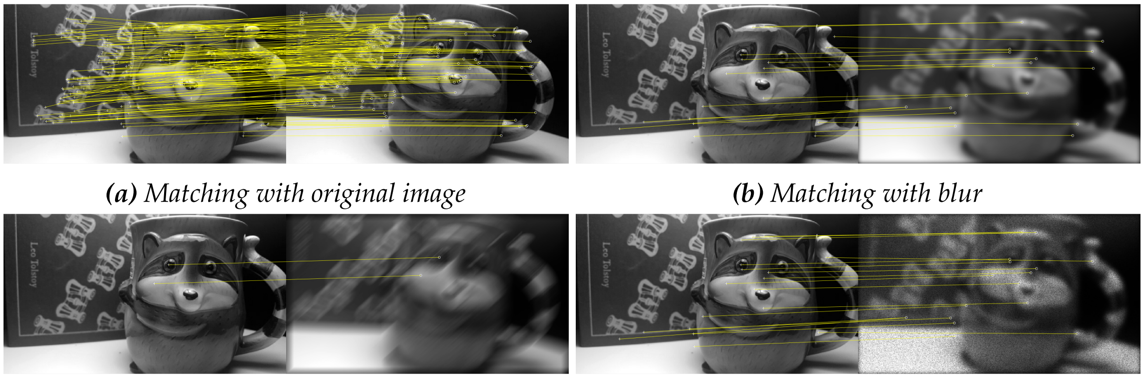
In addition to the photometric and geometric invariances, feature correspondences play an important role in batch geometric estimations problems. For instance, minimizing the reprojection error across multiple views using bundle adjustment is the most widely used approach to wide-baseline SFM [160, 372] and VSLAM applications. Using fixed correspondences in bundle adjustment is attractive for two reasons. Firstly, the reprojection error, though nonlinear, corresponds to a physical quantity in the world whose minimization is the Gold Standard [160] in contrast to algebraic linear alternatives. Secondly, it can be shown — at the least theoretically — that errors in feature tracking and localization over a long sequence tend to be normally distributed. Hence, minimizing the reprojection error provides the maximum likelihood solution [368, 369].

Nonetheless, feature descriptors have been designed to solve *sparse* correspondence problems, where the computational effort is fixated on two steps: (1) detecting locations of interest in the image characterized as salient [228, 323], and (2) extracting descriptors at these locations [373]. Interest point detection and descriptor extraction usually go hand in hand as the end goal is to maximize the distinctiveness of the extracted descriptors [260, 261]. When working with challenging data, the reliance on only a sparse collection of feature matches reduces the robustness as we demonstrate next.



(a) *Template* ***(b)*** *Blur* ***(c)*** *Blur and noise* ***(d)*** *Motion blur*

Figure 1.5: *Example of synthetically degraded imagery and extracted interest points.*



(a) *Matching with original image* ***(b)*** *Matching with blur*

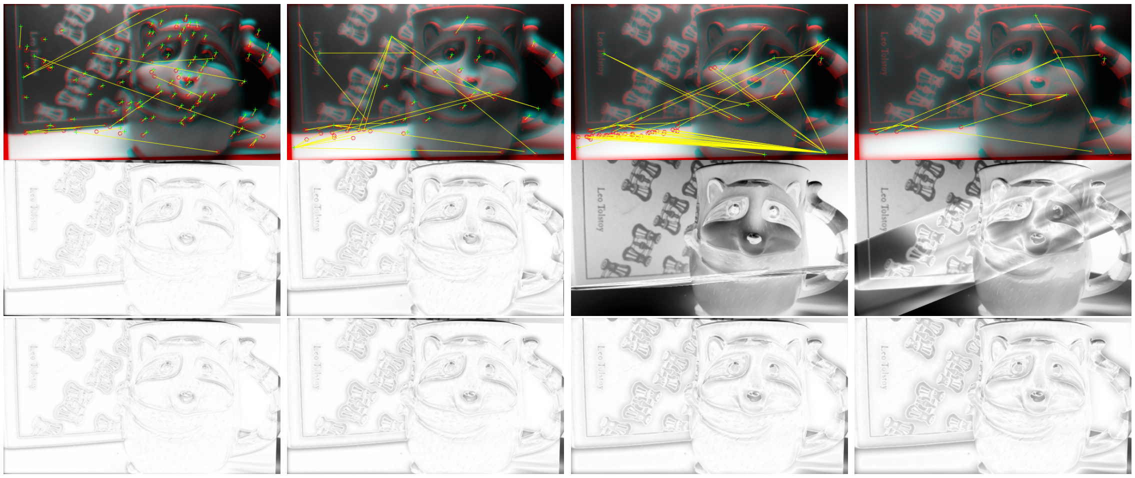
(c) *Matching with blur and noise* ***(d)*** *Matching with motion blur*

Figure 1.6: *Matching features with degraded imagery.*

Limitations of sparse features

In degraded imaging conditions, the difficulty of detecting and precisely localizing keypoints increases markedly. This is because the process of interest point detection examines only local areas in the image in isolation from the big picture. Consequently, keypoint detection algorithms make implicit assumptions about the adequacy of the signal-to-noise ratio in the local neighborhood for reliable for interest point detection [310, 323]. This is illustrated in Fig. 1.5, where the number and quality of the extracted interest point drops dramatically as the frequency content in the image degrades. In addition to degradations in the imaging quality, highly textured areas in the image such as corners and edges may not be readily available. In fact, in the majority of applications, most of the image is composed of areas of low texture while distinctive image points are improbable [228].

The quality of correspondences using standard sparse feature matching methods [37, 234] is shown in Fig. 1.6. The figure displays only "reliable" matches based on the descriptor



(a) $\sigma = 3$ ***(b)*** $\sigma = 5$ ***(c)*** $\sigma = 7$ ***(d)*** $\sigma = 8$

Figure 1.7: *Comparison of alignment performance between feature-based matching and direct intensity-based alignment as a function of Gaussian image blur. The image is rotated in-plane with* 5° *and translated along the* u*- and* v*-axis by* $(1.9,\ 2.1)$ *pixels respectively. Top row shows the matched SURF features [37]. Middle row shows the alignment error using MLESAC [367] with the matched features as input, where darker pixels indicate larger error. Bottom row is the performance of intensity-based direct alignment using the Lucas and Kanade algorithm [235]. By* $\sigma = 7$*, feature extraction and matching becomes unreliable and pose estimation fails. However, direct methods maintain a good accuracy.*

matching ratio test [37]. At a first glance, it appears that some matches are correct indicating that feature descriptors are distinctive enough despite the considerable degradation in image quality. This in fact is true. However, estimating the geometry relies on the pixel coordinates of the interest points, and not the feature descriptors. As we can see in Fig. 1.7, the accuracy of interest point localization deteriorates rapidly as a function of image degradations and quickly becomes insufficient for precise pose estimation tasks.

Direct pose estimation

By not restricting ourselves to a sparse selection of keypoints and instead using the whole image data directly, a robust solution for the correspondence problem is possible. As demonstrated at the bottom row of Fig. 1.7, using all pixels in the image pulls the estimation to the correct solution even though the frequency content in the image is reduced. This behavior has been described as the "locking property" in the literature [181].

Although it is possible to improve the robustness of correspondence estimation by simply using all image data, the approach is not without limitations. The main limitation of the direct approach is the assumption of brightness conservation, or *brightness constancy*[1] between the matched images. This failure mode is illustrated in Fig. 1.8 and it is a great obstacle to a wide spread use of direct methods in many applications.

Due to the importance of handling appearance variations, several methods have been proposed [18, 42, 81, 83, 130, 139, 164, 266, 304, 326, 334, 383], which we will review in detail in this dissertation. Most approaches to handling illumination variations, however, rely on implicit or explicit assumptions about the content of the scene or the type of illumination change. These modeling assumptions are by definition difficult to craft correctly and are not always satisfied, thereby limiting applications to specific instances where the assumptions can be met.

Improving direct methods using feature-based techniques

At least conceptually, the use of direct methods for pose estimation in challenging imaging conditions such as the ones shown in Figs. 1.1 and 1.2 appears to be a better solution than relying on sparse feature matches, especially under small motions. This is for two reasons: (i) the sheer redundancy possible when using the image data directly makes direct algorithms naturally more robust, and (ii) erroneous correspondences have less effect on the precision of estimates due to the iterative nature of the estimation process. Put simply, there is a chance to correct the correspondences during the coarse of the optimization.

Notwithstanding, there are two main limitations that we must first address for direct methods to compete with the feature-based pipeline in robotic application domains. The first is enhancing robustness against violations of the brightness constancy assumption that are commonly and frequently encountered in real data. The second is enabling direct methods to use multiple views for drift reduction without stringent assumptions on the scene structure or the type of camera motion.

In this work, we address the limitations of direct methods using well-established techniques from the feature-based pipeline. Firstly, we make use of local feature descriptors as a nonparametric means to handle violations of the brightness constancy assumption. The use of feature descriptors in direct alignment is not only robust, but makes little assumptions about the environment and lighting conditions.

[1] The term brightness (lightness constancy) was first coined in Psychology research and defined as the invariance of the perception of brightness of a constant reflectance surface as a function of varying illumination [131]. The canonical experiment establishing the existence of brightness constancy in the human visual system was conducted by Katz [197] in 1911 (as cited in [131]). This hypothesis has been verified in biology and has been shown to manifest at the early stages of processing in the visual cortex [178, 237, 308].

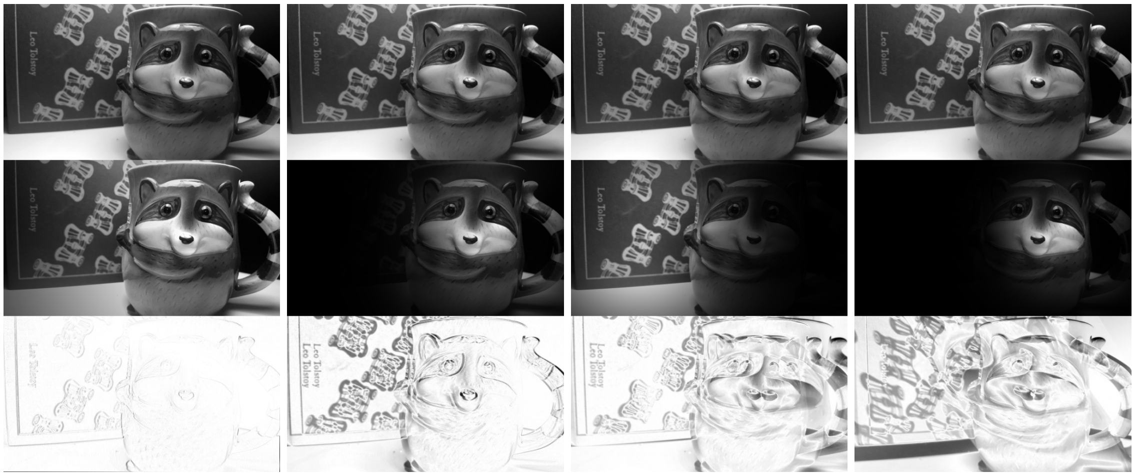


Figure 1.8: Limitations of direct methods for establishing correspondences due to changes in brightness. Images are augmented with a nonlinear form of appearance variations (radial point light source with contrast stretching). Intensity of appearance changes increases from left–right. Even at moderate nonlinear intensity change shown at the leftmost column direct alignment error is noticeable. Although intensity change at the leftmost column appears small, a closer look at the distribution of intensities shown in Fig. 1.9 indicates otherwise. For this example, we use a Lucas and Kanade algorithm with an affine motion model.

Secondly, we adapt the powerful bundle adjustment framework to the photometric domain, where instead of minimizing the reprojection error, we maximize the photometric consistency while jointly refining the parameters of the structure and motion.

In summary, this work aims to answer the following two main research questions:

1. If feature descriptors are powerful tools for correspondence estimation, can we still utilize their power without limiting their application to sparse interest points?

2. Can we go beyond local frame–frame estimation and use information from multiple images for geometric estimation directly similar to the state-of-the-art geometric estimation methods based on feature correspondences? Namely, can we perform a joint optimization akin to bundle adjustment with applications to VSLAM in unconstrained environments?

1.3 Thesis Statement

In this dissertation, we advocate the following statements

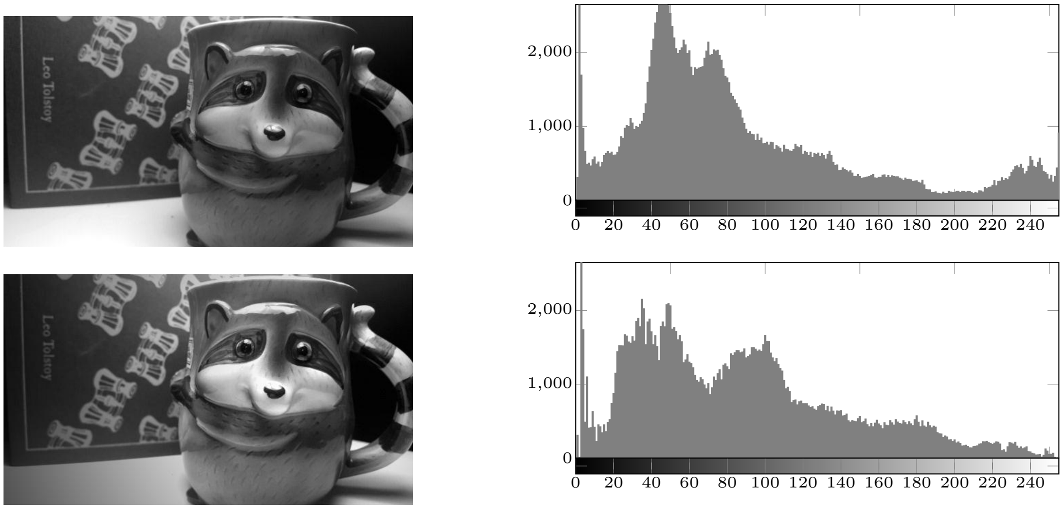


Figure 1.9: *Variations in appearance may appear negligible at a first glance. Nonetheless, a closer inspection of the distribution of intensities indicates significant differences that cannot be accounted for by relying on the brightness constancy only.*

(S1) Using as much of the image directly is a robust and efficient approach to pose estimation.

(S2) The approach can be made more robust by using densely evaluated feature descriptors in lieu of raw intensities.

(S3) The approach can be extended to work with multiple frames for tasks such as the simultaneous refinement of motion and structure (without imposing special requirements on the scene structure or camera motion), which to date has been limited to sparse keypoints.

The first statement, *S1*, will be demonstrated throughout this dissertation, particularly in Chapter 3. The second, *S2*, will be shown in Chapters 5 and 6. Finally, *S3*, will be shown in Chapter 7.

1.4 Contributions & Organization

This dissertation is organized as follows:

- In Chapter 2 (Page 13), we provide a review of preliminaries, background and some related literature. Chapter 2 also introduces notations used throughout this dissertation.

- Chapter 3 (Page 59) details the development and evaluation of a direct visual odometry approach from stereo data. The work verifies that using image data directly is a feasible and a robust approach to obtaining accurate pose information from a stream of video. In contrast to the state-of-the-art, our formulation makes use of disparities, which avoids the difficulties associated with using triangulated 3D points from stereo. This is akin to using inverse depth in vision-based SLAM [70].

- Chapter 4 (Page 75) is a continuation of the previous chapter, where we study important implementation details common to most direct visual odometry algorithms. These include the regularity of image interpolation, the numerical scheme used to estimate image gradients, as well as the effect of the number and distribution of pixels on the solution's accuracy. The evaluation is carried about using a range of synthetic and real benchmarks.

- Chapter 5 (Page 105) introduces our novel binary descriptor suitable for direct nonlinear least-squares optimization. The experimental portion of this chapter answers questions of theoretical nature. Hence, synthetic data, with known ground truth is used to understand the performance and limitations of the descriptor in comparison to other developments in the literature.

- Chapter 6 (Page 121) evaluates the idea of using densely extracted feature descriptors to tackle pose estimation tasks in challenging environments. Performance is demonstrated on two pose estimation problems: (1) planar template tracking under sudden and drastic changes in illumination, and (2) visual odometry in poorly lit subterranean mines. For planar template tracking, we augment a multi-channel formulation of LK with our bit-planes descriptor to demonstrate robust and faster than real-time performance. The novel visual odometry algorithm developed in this chapter is based on the one presented in Chapter 3, but uses bit-planes (and other local descriptors) to enhance robustness. In addition to using bit-planes, we also demonstrate the performance of several descriptors for a fair comparison.

- Chapter 7 (Page 151) goes beyond frame–frame local pose estimation and tackles the challenging problem of VSLAM. Here, we develop a correspondence-free multiview pose and structure refinement framework using image data directly. We call our framework *bundle adjustment without correspondences* as it shares many aspects of the well-known minimization of the reprojection error using geometric bundle adjustment. In contrast to previous work, — where the parameters of motion and structure are refined in an alternating fashion — the proposed framework demonstrates the feasibility of refining the parameters of pose and structure *simultaneously*. We also show that our framework is capable of improving on VSLAM results obtained with geometric bundle adjustment and loop closure combined.

- Finally, in Chapter 8 (Page 175) we present the conclusions of this dissertation, and in Chapter 9 (Page 179) we provide a summary of future research directions.

CHAPTER 2

Background and Related Work

> The farther backward you can look, the farther forward you are likely to see.
>
> Winston Churchill

Contents

This chapter provides a high-level overview of background materials and introduces the notations used in this dissertation.

2.1 Notation

Scalars will be denoted by unadorned symbols (*e.g.* s), vectors will be denoted with lowercase bold typeface (*e.g.* $\mathbf{v}$), while matrices will be denoted with an uppercase bold typeface (*e.g.* $\mathbf{M}$). Vectors are assumed to be column-vectors, their transpose is denoted with $\mathbf{v}^\top$.

The Euclidean norm (L_2 norm) of a vector will be denoted by $\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{d} \mathbf{v}_i}$, where $\mathbf{v}_i$ denotes the i^{th} element of the vector whose dimension is d, *i.e.* $\mathbf{v} \in \mathbb{R}^d$.

The parameters we seek to estimate will be denoted with the Greek alphabet. They are usually p-vectors (*e.g.* $\boldsymbol{\theta} \in \mathbb{R}^p$).

Function will be denoted similar to vectors, for example the notation:

$$\mathbf{f} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^k$$
$$\mathbf{f}(\mathbf{a}, \mathbf{b}) = \mathbf{c}$$

denotes a function $\mathbf{f}$ whose domain is the Cartesian product of an m-dimensional space with an n-dimensional space, and whose range (output) is k-dimensional. An exception is the camera projection function, which will be denote with $\pi(\cdot)$ as is commonly used in the literature. The inverse of the camera projection will be denoted with $\pi^{-1}(\cdot,\cdot)$, which in general depends on the intrinsic parameters of the camera and the scene depth.

The symbol $\mathbf{K}$ will be reserved for the 3×3 upper-triangular camera intrinsic matrix.

2.2 Lie Groups

Lie groups, and their algebras, are a large field with many applications in the mathematics and physics. They also play an important role in geometric optimization problems in Robotics and Computer Vision. Most relevant to this work is the use of the exponential map to represent the parameters of camera motion. Which we use to represent the 6DOF rigid-body pose in Chapters 3, 6 and 7 as well as the 8DOF parameters of a planar homography warp in Chapters 5 and 6.

We start by defining what a Lie group is, then provide examples and intuition.[1]

Definition 2.2.1 (Lie group)**.** A Lie group $\mathcal{G}$ is a continuous and smooth manifold. Multiplication and inverse maps are also smooth.

To understand the definition of Lie group, we must first define what a group is, and what a manifold is.

[1] The topic of Lie groups and their algebra is expansive and has applications in many areas of mathematics. We will present simplified definitions for ease of presentation and relevance to our application. Additional references are provided for a more rigorous treatment.

Definition 2.2.2 (Group)**.** A group is composed of two entities, a nonempty set $\mathcal{G}$ and an operator $\bullet$ such that for $a, b, c \in \mathcal{G}$ the following group axioms are satisfied:

(G1) **Associativity**
The group action must be associative $a \bullet (b \bullet c) = (a \bullet b) \bullet c$;

(G2) **Identity element**
There exists a unique $e \in \mathcal{G}$ such that $e \bullet a = a \bullet e = a$;

(G3) **Inverse element**
There exists $a^{-1} \in \mathcal{G}$ such that $a \bullet a^{-1} = a^{-1} \bullet a = e$, where e is the identity ;

(G4) **Closure**
The group action must remain in the group, *i.e.* $a \bullet b \in \mathcal{G}$ is also in $\mathcal{G}$.

Commutativity is a not a group requirement, but if the group action (operator) commutes, then the group may be called *abelian*.

Groups are commonly encountered in physics and mathematics. For instance, the set of integers $\mathbb{Z}$ under addition is a group denoted by $(\mathbb{Z}, +)$. Its identity element is 0. As addition commutes, it is also an abelian group. The set of integers excluding zero ($\mathbb{Z}/\left\{0\right\}$) under multiplication, however, is not a group because the inverse element is not part of the integers set.

Beyond numbers, the set of all $n \times n$ invertible matrices using real (or complex) coefficients under multiplication is also a group. Its identity element is the $n \times n$ identity matrix, $\mathbf{I}_n$. The group is called the *general linear group* and is denoted by $GL(n, \mathbb{R})$ (or $GL(n, \mathbb{C})$). Since we work with matrices of real coefficients, we will simply use $GL(n)$ to mean $GL(n, \mathbb{R})$.[2] Formally, the set of matrices in the general linear group is defined by

$$GL(n) \coloneqq \Big\{ \mathbf{G} \in \mathbb{R}^{n \times n} \ : \ \det(\mathbf{G}) \neq 0 \Big\} . \tag{2.1}$$

A subset of matrices in $GL(n)$ whose determinant is unity is called the special linear group of dimension n and denoted with $SL(n)$, *i.e.*

$$SL(n) \coloneqq \Big\{ \mathbf{S} \in \mathbb{R}^{n \times n} \ : \ \det(\mathbf{S}) = 1 \Big\} . \tag{2.2}$$

As we shall see later, $SL(n)$ is of particular interest to us where we use it to represent

[2] Lie groups of complex vs. real coefficients behave similarly, but their dimensionality differs. This distinction is irrelevant to our use of Lie Groups and their Algebras.

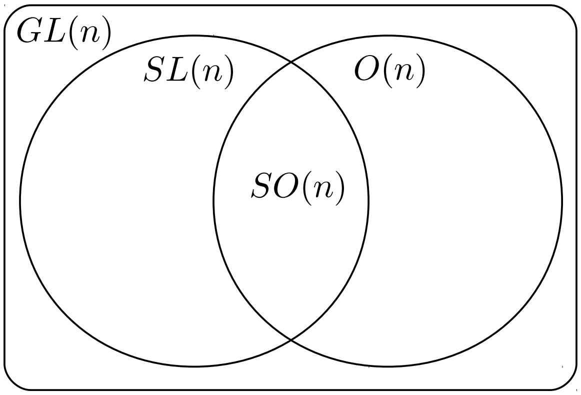


Figure 2.1: *Relationship between the rotation group $SO(n)$ and other Lie groups.*

plane-induced homographies.

Another subset of $GL(n)$ is the set of orthogonal matrices called the orthogonal group and denoted with $O(n)$. Matrices in this set must obey

$$O(n) \coloneqq \left\{ \mathbf{O} \in \mathbb{R}^{n \times n} \ : \ \mathbf{O}\mathbf{O}^\top = \mathbf{O}^\top\mathbf{O} = \mathbf{I}_{n \times n} \right\}, \tag{2.3}$$

where $\mathbf{I}_{n \times n}$ is the $n \times n$ identity matrix. The definition of the orthogonal group of matrices implies matrices with unit determinant, but has no restriction on the sign of the determinant. Hence, the group contains both: rotation and reflection transforms in n-space.

The intersection of $SL(n)$ and $O(n)$ gives rise to the special orthogonal group $SO(n)$. This is set of orthogonal matrices whose determinate is $+1$. In other words, the set of rotation matrices in n-dimensions. The relationship between these groups is illustrated in Fig. 2.1.[3]

The second part of a Lie Group's definition is the requirement of being a smooth manifold. A mathematical manifold is defined as follows:

Definition 2.2.3 (Manifold)**.** An n-dimensional manifold is a topological space that locally behaves as an n-dimensional Euclidean space.

Informally, a manifold is a nonlinear structure, but it behaves locally as if it was linear as illustrated in Fig. 2.3. The linearity part is due to the Euclidean structure as every element in the Euclidean space is a linear combination of a set of finite basis. This local linearity is particularly important for optimization problems.

The most intuitive example of a manifold is the surface of the Earth, which for the purpose

[3] The analog to $SL(n)$ and $O(n)$ with complex coefficients is $UL(n)$ and $U(n)$ with transposition replaced with complex conjugation, where U is a short for unitary.

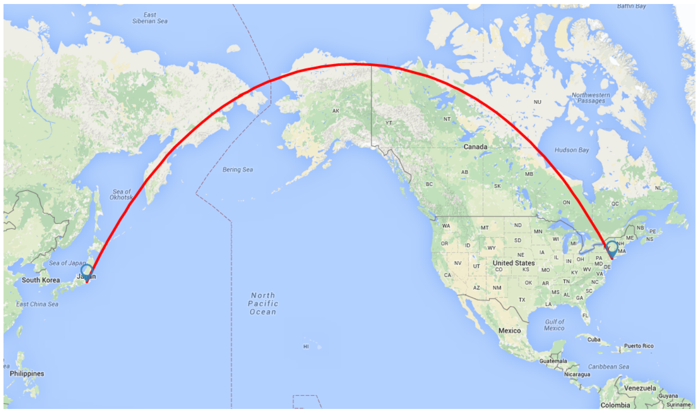


Figure 2.2: *The shortest distance between New York and Tokyo is a great arc. Information was obtained from* https://www.freemaptools.com/ *and Google. Interestingly, most of the flight path is above land.*

of this discussion, we approximate as a large sphere. For most day-to-day activities, say a trip to the grocery store, the shortest distance between two points is a straight line.[4] However, if one is to travel a longer distance, the straight line approximation of the shortest distance becomes poor. Instead, the shortest distance is now measured with a great arc as shown in Fig. 2.2. This generalization of the notion of a straight line to curved spaces is commonly known as the geodesic distance.

2.2.1 Lie Algebra

Every Lie group has associated a Lie algebra defined as:

Definition 2.2.4 (Lie algebra)**.** A Lie algebra is a vector space with an operator called the *bracket*, or the matrix commutator, and is denoted by $[\cdot, \cdot]$. Given $\mathbf{a}, \mathbf{b}, \mathbf{c}$ in the vector space, the following must hold:

- **Skew-symmetry** $[\mathbf{a}, \mathbf{a}] = 0$, which also implies $[\mathbf{a}, \mathbf{b}] = -[\mathbf{b}, \mathbf{a}]$
- **Jacobi identity** $\left[\mathbf{a}, [\mathbf{b}, \mathbf{c}]\right] = \left[\mathbf{b}, [\mathbf{c}, \mathbf{a}]\right] = \left[\mathbf{c}\, [\mathbf{a}, \mathbf{b}]\right]$

It is common to denote the Lie algebra with lower case Fraktur, *e.g.* the Lie algebra of $GL(n)$ is denoted $\mathfrak{gl}(n)$.

The basis of the Lie algebra are commonly known as the *generators*. For example, consider

[4] If one is not restricted by the topology of road networks.

$SO(3)$ the group of rotations in 3-space. Its generators are given by:

$$\mathbf{G}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \ \mathbf{G}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \text{ and } \mathbf{G}_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{2.4}$$

In fact, the generators of $\mathfrak{so}(3)$ in Eq. (2.4) are derivatives of the infinitesimal rotations of 3-space about the basis vector. Given any vector $\boldsymbol{\omega} \in \mathbb{R}^3$, we have

$$\sum_{i=1}^{3} \boldsymbol{\omega}\mathbf{G}_i \in \mathfrak{so}(3). \tag{2.5}$$

A Lie algebra is the tangent space of its associated group at the identity element. For $\mathfrak{so}(3)$, they are the skew-symmetric matrices generated from 3-vectors, *i.e.* $[\boldsymbol{\omega}]_\times \in \mathfrak{so}(3)$, where for $\boldsymbol{\omega} = \left(\omega_x,\ \omega_y,\ \omega_z\right)^\top$,

$$[\boldsymbol{\omega}]_\times = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \in \mathfrak{so}(3). \tag{2.6}$$

For convenience, however, we will simply say $\boldsymbol{\omega} \in \mathfrak{so}(3)$.

An important concept is the exponential map (and its inverse: the log map), which allows us to transition from the group to its algebra (the tangent space about the identity). The exponential map is reviewed in the next section.

2.2.2 The exponential map

The exponential map is the tool to map elements from the tangent space (the Lie algebra) to the Lie group. For instance, consider 3-space, for $\boldsymbol{\omega} \in \mathbb{R}^3$ we have

$$\exp\left([\boldsymbol{\omega}]_\times\right) \in SO(3). \tag{2.7}$$

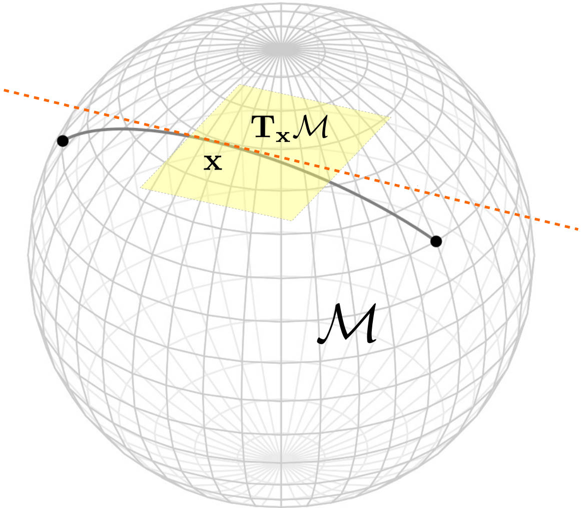


Figure 2.3: *Illustration of a manifold $\mathcal{M}$ and its tangent space $\mathbf{T_x}\mathcal{M}$ at a point $\mathbf{x}$. The line (vector) through $\mathbf{x}$ approximates an arc on the sphere only locally.*

The matrix exponential of an $n \times n$ real (or complex) matrix $\mathbf{M}$ is defined—analogously to the usual scalars—via the power series given by:

$$\exp(\mathbf{M}) = \sum_{i=0}^{\infty} \frac{1}{i!}\mathbf{M}^i \tag{2.8}$$

and follows the usual exponentiation rules. In general, however, the power series in Eq. (2.8) is divergent and must be approximated. Padé approximation is usually the method of choice [31].[5]

Of particular interest to us are two Lie gruops: the first is the Special Euclidean group in three dimensions $SE(3)$, or the group of rigid-body transformations. The second is the special linear group $SL(3)$, which we used to represent plane-induced homographies. Both groups are briefly reviewed next.

[5] This is implemented in MATLAB using the function `expm`.

$$\begin{tikzcd}
{[\boldsymbol{\omega}]_\times \in \mathfrak{so}(3)} \arrow[rr, bend left, "\exp\left([\boldsymbol{\omega}]_\times\right)"] & & \mathbf{R} \in SO(3) \arrow[ll, bend left, "\ln\left(\mathbf{R}\right)"]
\end{tikzcd}$$

Figure 2.4: *Action of the exponential and log maps*

2.2.3 The group of rigid-body transformations $SE(3)$

A rigid-body transformation (pose) in 3-space is composed of a rotation matrix and a translation vector and takes the form

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \in SE(3). \tag{2.9}$$

The translation part of the pose matrix bears no difficulty in optimization problems as it is Euclidean, *i.e.* linear. Metrics on the rotation space are, however, nonlinear, and their use in optimization must be approached with care [162, 179].

Rotations in 3-space have three degrees-of-freedom. Yet, their matrix representation is composed of nine entries. We could optimize over rotations using the nine elements of the rotation matrix. But, this is impractical and inadvisable as: (1) it increases the dimensionality of the state vector, thus making the problem more difficult, and (2) enforcing the rotation matrix constraints (orthogonality with positive unit determinant) is a nonlinear operation; normalization after the fact usually does not produce good results.

Quaternions are a popular tool to represent rotations in robotics [142], but care must be taken to ensure that the quaternion maintains a unit norm for it to be a valid representation of a 3D rotation.[6]

The Lie group $SE(3)$ and its algebra $\mathfrak{se}(3)$ provide a convenient way to represent rigid-body transformations. For instance, commonly performed operations such as composition

[6] When normalized to unit vectors, quaternions can be represented using the 3-sphere Lie Group: the Special Unitary Group $SU(2)$ which is isomorphic to unit quaternion.

of transformations, interpolation, inversion and uncertainty estimation can be elegantly performed using $SE(3)$ and its algebra.

Formally, $SE(3)$ is the product of two manifolds $SO(3)$ and $\mathbb{R}^3$, usually denoted with

$$SE(3) \coloneqq SO(3) \times \mathbb{R}^3.$$

The generators of $\mathfrak{se}(3)$ are the following six 4×4 matrices:

$$\mathbf{G}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \; \mathbf{G}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}, \; \mathbf{G}_3 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.10}$$

$$\mathbf{G}_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \; \mathbf{G}_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \text{ and } \mathbf{G}_6 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{2.11}$$

such that any rigid-body transformation may be represented as:

$$\mathbf{T}(\boldsymbol{\theta}) = \exp\left(\sum_{i=1}^{6} \boldsymbol{\theta}_i \mathbf{G}_i\right), \tag{2.12}$$

where $\boldsymbol{\theta} \in \mathbb{R}^6$.

Closed-form solution of the exponential and log maps for $SE(3)$

Given the vector $\boldsymbol{\theta} = (\omega,\ \boldsymbol{\nu})^\top \in \mathbb{R}^6$, we first define the hat-operator as

$$\hat{\boldsymbol{\theta}} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 & \nu_1 \\ \omega_3 & 0 & -\omega_1 & \nu_2 \\ -\omega_2 & \omega_1 & 0 & \nu_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} [\boldsymbol{\omega}]_\times & \boldsymbol{\nu} \\ \mathbf{0}^\top & 0 \end{pmatrix}. \tag{2.13}$$

The matrix exponential to map from the vectorial representation to a rigid-body pose is given by

$$\exp\left(\hat{\boldsymbol{\theta}}\right) = \begin{pmatrix} \exp\left([\boldsymbol{\omega}]_\times\right) & \mathbf{V}\boldsymbol{\nu} \\ \mathbf{0}^\top & 1 \end{pmatrix}, \tag{2.14}$$

where

$$\exp\left([\boldsymbol{\omega}]_\times\right) = \mathbf{I} + \frac{\sin\phi}{\phi}\,[\boldsymbol{\omega}]_\times + \frac{1-\cos\phi}{\phi^2}\,[\boldsymbol{\omega}]_\times^2\,, \tag{2.15}$$

with

$$\phi = \|\boldsymbol{\omega}\| \tag{2.16}$$

and $\mathbf{I}$ denotes the 3×3 identity matrix. The matrix $\mathbf{V} \in \mathbb{R}^{3\times3}$ takes the form

$$\mathbf{V} = \mathbf{I} + \frac{1-\cos\phi}{\phi^2}\,[\boldsymbol{\omega}]_\times + \frac{\phi-\sin\phi}{\phi^3}\,[\boldsymbol{\omega}]_\times^2 \tag{2.17}$$

We note, however, if ϕ is near zero then the rotation axis is undefined. Hence, this case must be handled specially such that $\exp\left([\boldsymbol{\omega}]_\times\right) = \mathbf{I}$.

Similarly, the log map has a closed-form. Given a rigid-body transformation matrix of the form

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 0 \end{pmatrix}, \tag{2.18}$$

let

$$\theta = \cos^{-1}((\operatorname{trace}\mathbf{R} - 1)/2), \tag{2.19}$$

then

$$\log\mathbf{T} = \begin{pmatrix} \log\mathbf{R} & \mathbf{V}^{-1}\mathbf{t} \\ \mathbf{0}^\top & 0 \end{pmatrix}, \tag{2.20}$$

where

$$\log \mathbf{R} = \frac{\theta}{2\sin\theta}\left(\mathbf{R} - \mathbf{R}^{\top}\right), \tag{2.21}$$

and

$$\mathbf{V}^{-1} = \mathbf{I} - \frac{1}{2}\log \mathbf{R} + \frac{1}{\theta^2}\left(1 - \frac{a}{b}\right)\log \mathbf{R}^2, \tag{2.22}$$

with

$$a = \frac{\sin\theta}{\theta}, \text{ and } b = 2\frac{1-\cos\theta}{\theta^2}. \tag{2.23}$$

Finally, extracting the 6-vector from the log map is performed using the vee-operator, which is the inverse of the hat-operator defined in Eq. (2.13). Similar to the exponential map, if θ is near zero, then $\mathbf{V}^{-1} = \mathbf{I}$ and the rotation part—$\boldsymbol{\omega}$—is the zero 3-vector.

2.2.4 The Special Linear Group for Parameterizing Homography $SL(3)$

The imaging model of planes under perspective can be described by a 3×3 projective transformation known as a homography [82]. The homography encodes the eight degrees-of-freedom describing the plane motion and its decomposition can be used to estimate the pose of the camera given known calibration parameters [242]. Homography estimation is not only important for describing the motion of planar targets, it also arises when the motion of the camera is purely rotational about its center of projection. Applications of homography estimation also include rectification [163, 229] and visual reconstruction using mosiacs [214, 354].

Due to common applications of homographies in geometric estimation problems, several parameterizations have been proposed [30]. An elegant parametrization, however, is using the $SL(3)$ group. Recall that the Homography is defined up to scale, and if this scale is chosen such that the determinant of the matrix is unity, then the $SL(3)$ parameterization naturally arises.

The generators for $SL(3)$ are the following eight 3×3 matrices:

$$\mathbf{G}_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{G}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{G}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{G}_4 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (2.24)$$

$$\mathbf{G}_5 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{G}_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \mathbf{G}_7 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{G}_8 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (2.25)$$

Unlike the group of rigid-body transformations, the exponential map from $\mathfrak{sl}(3)$ to the group elements of homographies $SL(3)$ (and its inverse: the logarithm map) has no closed-form and must be approximated [267].

2.2.5 Additional details

Gallier [134] provides an in-depth treatment of Lie Groups and their applications in differential geometry. In compute vision, Lee [218] presents solutions to geometric problems using the Lie Group theory. In particular, they present solutions for pose estimation and simultaneous registration of multiple three-dimensional point clouds. Bregler and Malik [54] demonstrate the utility of Lie Groups for tracking high DOF objects.

Grassia [142] convincingly argues the benefits of the exponential map to represent rotations in computer graphics. In robotics, Murray et al. [277] provide a concise summary and closed-form formulae for commonly used exponential maps.

The topic is in fact very useful and there exists many well-written notes and books, which this summary derives from [45, 183, 204]

2.3 Camera Projection

A simple definition of a camera is a device that captures the projection of the light field onto a 2D plane. A more rigorous definition was recently provided by Ponce [302]. In this work, we are interested in the most commonly used camera, the *pinhole camera* illustrated in Fig. 2.5.

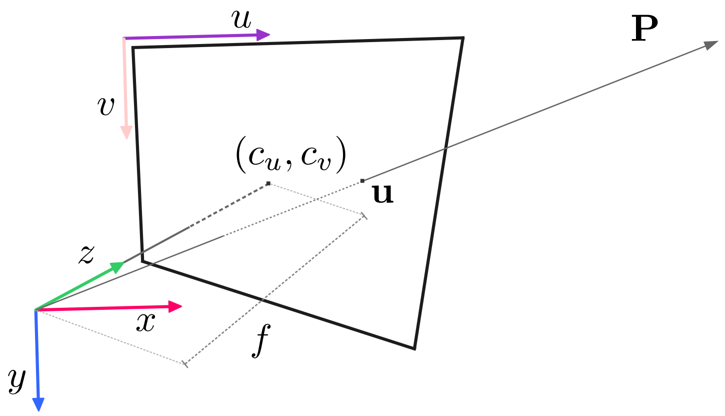


Figure 2.5: Pinhole projection model.

The pinhole projection, assuming square pixels, is characterized by the 3×3 projection matrix which takes the form

$$\mathbf{K} = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.26}$$

where f_u and f_v are the focal lengths of the u- and v-axis respectively, and the principle point is denoted with (c_u, c_v). The camera is placed in a right-handed coordinate system as shown in Fig. 2.5, where the action of projection of a 3D scene point onto the image plane is given by the projection function

$$\pi : \mathbb{R}^3 \to \mathbb{R}^2 \tag{2.27}$$

$$\pi(\mathbf{P}) = \mathbf{n}(\mathbf{KP}), \tag{2.28}$$

where $\mathbf{n}(\cdot)$ denotes de-homogeneouzing by dividing the first two coordinates of the input by the third, *i.e.*: $\mathbf{n}([a, b, c]^\top) = \left[{}^a\!/\!_c, {}^b\!/\!_c\right]^\top$.

When $\mathbf{K}$ is known, the camera is said to be calibrated. In which case, the camera behaves as an angle measurement device. Every point on the image plane $\mathbf{u} = [u, v, 1]^\top$ makes an angle with the center of projection. The ray emitting from the center of projection and piercing throw the pixel $\mathbf{u}$ is given by

$$\hat{\mathbf{u}} = \mathbf{K}^{-1}\mathbf{u}. \tag{2.29}$$

This ray intersects the image plane at distance f (focal length) from the center of projection.

Given a depth measurement d associated with $\hat{\mathbf{u}}$, we may obtain the Euclidean position of the point in space using

$$\mathbf{P} = d\hat{\mathbf{u}}. \tag{2.30}$$

This action defines the inverse camera projection given by

$$\pi^{-1} : \mathbb{P}^2 \times \mathbb{R} \to \mathbb{R}^3 \tag{2.31}$$

$$\pi^{-1}(\mathbf{u}, d) = d\mathbf{K}^{-1}\mathbf{u}, \tag{2.32}$$

where $\mathbf{K}^{-1}$ can be explicitly obtained as

$$\mathbf{K}^{-1} = \begin{pmatrix} {}^{1}\!/\!{}_{f_u} & 0 & -{}^{c_u}\!/\!{}_{f_u} \\ 0 & {}^{1}\!/\!{}_{f_v} & -{}^{c_v}\!/\!{}_{f_v} \\ 0 & 0 & 1 \end{pmatrix} . \tag{2.33}$$

Faugeras et al. [119, ch. 4], and Hartley and Zisserman [160, ch. 6] provide an in-depth treatment and discussion of the pinhole projection model as well as other commonly encountered projection models and nonlinear distortion due to optical lenses.

2.4 Nonlinear Least-Squares

Given an objective, and some limited information, optimization answers the question: what is the best action to take?

For all, but a subset of problems known as convex, general optimization remains a difficult problem [52, 289]. A subset of convex optimization problems is known as (nonlinear) least-squares. Least-squares problem posses a special structure that makes their solution easier and more reliable than general purpose optimization problems.

Formally, given m smooth vector-valued functions $r_i : \mathbb{R}^n \to \mathbb{R}$, parameterized by the vector $\boldsymbol{\theta} \in \mathbb{R}^n$, we seek to find a local minimum[7] of the sum of the residuals

$$f(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{m} r_i^2(\boldsymbol{\theta}). \tag{2.34}$$

[7] If one seeks a maximum, we simply negate the objective

It is easier to proceed by stacking the residuals in a vector $\mathbf{r} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, which takes the form

$$\mathbf{r} = \big(r_1(\boldsymbol{\theta}), r_2(\boldsymbol{\theta}), \ldots, r_n(\boldsymbol{\theta})\big)^\top \in \mathbb{R}^{m \times 1} \ . \tag{2.35}$$

The least-squares optimization objective in Eq. (2.34) can now be written in matrix form as

$$f(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{r}\|_2^2 \, . \tag{2.36}$$

The derivative of f is the $m \times n$ Jacobian matrix given by:

$$\mathbf{J}(\boldsymbol{\theta})_{\substack{\text{i=1,...,m} \\ \text{j=1,...,n}}} = \begin{pmatrix} \nabla r_1(\boldsymbol{\theta})^\top \\ \nabla r_2(\boldsymbol{\theta})^\top \\ \vdots \\ \nabla r_m(\boldsymbol{\theta})^\top \end{pmatrix} , \tag{2.37}$$

where ∇r_j is the gradient of the j^{th} residual. As for the objective f, its gradient and Hessian at $\boldsymbol{\theta}$ take the form:

$$\nabla f(\boldsymbol{\theta}) = \sum_{i=1}^{m} r_i(\boldsymbol{\theta}) \nabla r_i(\boldsymbol{\theta}) \tag{2.38}$$

$$= \mathbf{J}(\boldsymbol{\theta})^\top \mathbf{r}(\boldsymbol{\theta}) \tag{2.39}$$

$$\nabla^2 f(\boldsymbol{\theta}) = \sum_{i=1}^{m} \nabla r_i(\boldsymbol{\theta}) \nabla r_i(\boldsymbol{\theta})^\top + \sum_{i=1}^{m} r_i(\boldsymbol{\theta}) \nabla^2 r_i(\boldsymbol{\theta}) \tag{2.40}$$

$$= \mathbf{J}(\boldsymbol{\theta})^\top \mathbf{J}(\boldsymbol{\theta}) + \sum_{i=1}^{m} \nabla^2 r_i(\boldsymbol{\theta}). \tag{2.41}$$

In contrast to computing second-order partial derivatives, computing the first-order partial derivatives (the Jacobian) is relatively easy and inexpensive to perform. When an initialization close to the local minima is provided, the second order terms ($\nabla^2 r_i(\boldsymbol{\theta})$) are relatively small. Hence, the Jacobian term has a more significant contribution to the objective. The same situation arises when the residuals — $r_i(\boldsymbol{\theta})$ — are relatively small. Thus, the first-order approximation to the Hessian — $\mathbf{J}(\boldsymbol{\theta})\mathbf{J}(\boldsymbol{\theta})^\top$ — provides a good estimate to the full Hessian without having to perform difficult to compute second-order derivatives. This unique property to least-squares problems make their solution attractive in practice [289].

While relying only on first-order derivative is attractive (more efficient and better numerical conditioning [371]) there is one caveat. The solution of the linear system of equations required to obtain a parameter update is now sensitive to scaling. For instance, if the magnitude of some elements of the parameter vector significantly differ (larger or smaller) than the rest, then the solution may be biased towards element with the higher magnitude. If this is the case, appropriate pre-scaling schemes must be performed to ensure a good and a numerically stable solution [372].

When residuals are distributed according to a Normal distribution, the solution is known as the Maximum Likelihood Estimate (MLE). If a prior on the parameters is known, a Bayesian approach may be used. Under a Bayesian formulation, the optimal estimate of parameters is obtained such that posteriori known probability distribution of the parameters is maximized, *i.e.*:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, f\left(\boldsymbol{\theta} \mid \mathbf{z}\right), \tag{2.42}$$

where $f\left(\cdot|\cdot\right)$ is the probability distribution of the parameter vector given a vector of measurements $\mathbf{z} = (z_1, \ldots, z_m)^\top$.

The probability of the parameters given the measurements is often hard to compute, or unknown. Conversely, the probability of measurements given the parameters is easier to model. Using Bayes rule we may write the objective in Eq. (2.42) as

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \frac{f(\mathbf{z} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta})}{f(\mathbf{z})}. \tag{2.43}$$

The probability of the measurements $f(\mathbf{z})$ is often called the marginal distribution and does not depend on the parameters we seek to estimate. Hence, it may be neglected. Similarly, the probability of the parameters $f(\boldsymbol{\theta})$ is constant and usually known (this is the prior). If we assume that the distribution of the parameters is uniform, the objective can now be written as

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, f(\mathbf{z} \mid \boldsymbol{\theta}). \tag{2.44}$$

If we assume that the measurements are independent and identically distributed (iid), then Eq. (2.44) is equivalent to the more tractable form given by

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{i=1}^{m} f(z_i\, |\boldsymbol{\theta}). \tag{2.45}$$

Finally, for even moderate values of m, the multiplication of probabilities vanishes quickly.

A mathematically equivalent form, but numerically stable is to minimize the negative log-likelihood given by

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{m} - \log f\left(z_i \mid \boldsymbol{\theta}\right). \tag{2.46}$$

Now, given a form of the probability distribution of the measurements given the parameters. A common assumption is using a Gaussian distribution. Let $f(z_i \mid \boldsymbol{\theta}) \sim \mathcal{N}(\mu, \sigma)$, with $\mu = z_i - \mathbf{f}(\boldsymbol{\theta})$, where $\mathbf{f}_i(\boldsymbol{\theta})$ is a function that predicates the i^{th} measurements given the parameters. Then, minimizing the log-likelihood in Eq. (2.46) becomes:

$$\begin{aligned} \boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{m} \frac{1}{\sigma^2} \left(z_i - \mathbf{f}_i(\boldsymbol{\theta})\right)^2 & (2.47) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \mathbf{r}_i(\boldsymbol{\theta})^2, & (2.48) \end{aligned}$$

where $\mathbf{r}_i(\boldsymbol{\theta})$ is the i^{th} residual.

If a prior is known on the distribution of parameters, it can be easily integrated in to the optimization. For instance, in the case of camera pose estimation, the prior may enforce a smoothness or continuity of the estimation based on the previous estimates of the parameters (*e.g.* constant acceleration). Let this prior be also normally distributed, *i.e.*

$$f(\theta) \sim \mathcal{N}(\mu_{\boldsymbol{\theta}}, \sigma_{\boldsymbol{\theta}}), \tag{2.49}$$

then the MAP estimate takes the form

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{m} \left(\frac{\mathbf{r}_i(\boldsymbol{\theta})}{\sigma}\right)^2 + \frac{1}{2\sigma_{\boldsymbol{\theta}}^2} \left|\mu_{\boldsymbol{\theta}} - \boldsymbol{\theta}\right|^2. \tag{2.50}$$

Inspecting the form of Eq. (2.50), the optimal estimate of the parameters not only minimizes the sum of squared residuals, but in addition penalizes deviation from the known prior of the parameters.

Finally, the normal equations are formulated akin to how we formulated them previously in Eq. (2.52), but additionally take into account the prior information. The normal equations under the Bayesian formulation take the form:

$$\mathbf{J}(\boldsymbol{\theta})^\top \mathbf{J}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta} = -\mathbf{J}(\boldsymbol{\theta})^\top \mathbf{r}(\boldsymbol{\theta}) + \sigma_{\boldsymbol{\theta}}^{-2} \left(\mu_{\boldsymbol{\theta}} - \boldsymbol{\theta}\right). \tag{2.51}$$

In the next section we discuss the two most common algorithms to solving least-squares problem: Gauss-Newton and Levenberg-Marquardt.

2.4.1 Solving nonlinear least-squares problems

The two main stream algorithms to solving nonlinear least-squares problems are the Gauss-Newton algorithm (GN) and the Levenberg-Marquardt algorithm (LM) [222, 245][8].

Both GN and LM perform three common steps repeatedly until convergence:

- *Linearization*, where the first-order partial derivatives (the Jacobian) are computed at the current estimate of the parameter vector.
- *Solving a linear system* using an appropriate linear solver (such as QR, or Cholesky decomposition), and
- *Updating the parameters*, where the incremental update obtained by solving the linear system is added to the parameter vector prior to repeating the process.

The main difference between GN and LM is how the linearized system of equations is constructed. In GN, the parameter update are obtained by solving what is commonly known as the *normal equations*, which at the k^{th} iteration are given by

$$\mathbf{J}_k^\top \mathbf{J}_k \Delta\boldsymbol{\theta}_{\text{GN}} = -\mathbf{J}_k^\top \mathbf{r}_k. \tag{2.52}$$

Advantages of GN include:

1. No need to compute second-order derivatives;
2. If the GN approximation to the Hessian ($\mathbf{J}_k^\top \mathbf{J}_k$) is a good approximation of the true Hessian, then convergence is quadratic; and
3. When the gradient ∇f is non vanishing, and $\mathbf{J}$ is full rank, the GN solution is the descent direction for the objective f.

Advantages of GN come with strong conditions such as accuracy of the first-order approximation to the Hessian (small residual problem) and the Jacobian is full rank (not always

[8] Powell's Doglog is another trust-region based method that is gaining popularity in Robotics [309] and Computer Vision [231], but we did not find a measurable advantage of employing DogLog it over either GN (for image alignment and direct visual odometry) and LM (for geometric and photometric bundle adjustment problems). In fact, LM was shown to be superior in other optimization fields [331].

possible due to measurement noise and roundoff errors). The LM algorithm addresses some of these issues by solving an *augmented* version of the normal equations, which is given by

$$\left(\mathbf{J}_k^\top \mathbf{J} + \lambda_k \mathbf{I}\right) \Delta\boldsymbol{\theta}_{\text{LM}} = -\mathbf{J}_k^\top \mathbf{r}_k, \tag{2.53}$$

where $\lambda_k > 0$ is adjusted at every iteration to ensure a reduction in the objective [239, 289].

Least-squares problems are known to be sensitives to outliers. The breakdown point of an estimator is percentage of outlier data points it can handle before giving incorrect results. In the case of ordinary least-squares, the breakdown point is 0%. In other words, a single outlier measurement throws off the estimate.

An approach to addressing outliers based on M-Estimators from robust statistics is summarized in the next section.

2.4.2 Robust Estimation

There are multiple methods to address outlier and obtain more robust estimators [177, 404], of particular interest is the class of M-Estimators. According to Huber [177], the M- stands for maximum likelihood type.[9]

The idea of M-Estimators is conceptually simple. The reason that ordinary least-squares is so sensitive to outliers is the high influence outliers bring to the objective. If instead we use an influence function that does not quadratically increase as a function of the residuals absolute magnitude, we can down weight the influence of large residuals on the solution. The rationale is that unusually large residuals are most likely gross outlier measurements.

While one may use any nonnegative function as an influence function, influence functions are usually symmetric,[10] positive definite, with a unique minimum at zero. They of course must grow at a rate less than the quadratic. Given a vector of residuals $\mathbf{r} = (r_1, \dots, r_m)^\top$, M-estimators reduce the influence of outliers by minimizing the following objective, or

[9] Other robust estimators in statistics include [153]: A-estimators (asymptotic variance), D-estimators (minimum distance), P-estimators (Pitman), L-estimators (linear combination of order statistics), S-estimators (scale statistic), R-estimator (rank test) and W-estimator (weighted mean).

[10] The requirements here are not strict. For instance, asymmetric M-Estimators have been proposed by Allende et al. [16] designed to work with asymmetric distributions.

loss function

$$\sum_{i=1}^{m} \rho(r_i), \tag{2.54}$$

where the function ρ is chosen such that it grows at rate slower than least-squares. The M-estimator of a parameter vector $\boldsymbol{\theta}$ based on the function ρ is the solution to

$$\sum_{i=1}^{m} \psi(r_i) \frac{\partial r_i}{\partial \boldsymbol{\theta}}, \tag{2.55}$$

where the function ψ is the known as the *influence function* and is given by the derivative:

$$\psi(x) = \frac{\partial \rho(x)}{\partial x}. \tag{2.56}$$

If we let the $w(x)$ be the weight function given by

$$w(x) = \frac{\psi(x)}{x}, \tag{2.57}$$

then Eq. (2.56) may be written as:

$$\sum_{i=1}^{m} w(r_i^{(k-1)}) r_i^2, \tag{2.58}$$

where $r_i^{(k-1)}$ indicates the value of the residuals at the previous iteration $(k-1)$. The form of Eq. (2.56) is identical to ordinary least-squares, but is now weighted using the function w accounting for large residuals that are most likely outliers.

Multiple ρ functions have been proposed in the literature. There is, however, no single *best* function as the performance of the estimator is tied to the data, type of noise and the closeness of the initial estimates to the optimal value.

Two popular functions are the Huber and the Tukey influence functions. The Huber function, given a tuning constant k, takes the form

$$\rho(x) = \begin{cases} \frac{x^2}{2} & \text{if } |x| \leq k; \\ k\left(|x| - {}^{k}\!/_{2}\right) & \text{otherwise.} \end{cases} \tag{2.59}$$

$$\psi(x) = \begin{cases} x & \text{if } |x| \leq k; \\ k\,\text{signum}(x) & \text{otherwise.} \end{cases} \tag{2.60}$$

$$w(x) = \begin{cases} 1 & \text{if } |x| \leq k; \\ \frac{k}{|x|} & \text{otherwise.} \end{cases} \tag{2.61}$$

The tunning constant is usually selected to be $k = 1.345$ to achieve 95% asymptotic efficiency of the normal distribution.[11] Tukey's function, on the other hand, also known as Tukey bi-weight takes the form

$$\rho(x) = \begin{cases} \frac{t^2}{6}\left(1 - \left({}^{x}\!/_{t}\right)^2\right)^3 & \text{if } |x| \leq t; \\ \frac{t^2}{6} & \text{otherwise.} \end{cases} \tag{2.62}$$

$$\psi(x) = \begin{cases} x\left(1 - \left({}^{x}\!/_{t}\right)^2\right)^2 & \text{if } |x| \leq t; \\ 0 & \text{otherwise.} \end{cases} \tag{2.63}$$

$$w(x) = \begin{cases} \left(1 - \left({}^{x}\!/_{t}\right)^2\right)^2 & \text{if } |x| \leq t; \\ 0 & \text{otherwise.} \end{cases} \tag{2.64}$$

The tunning constant for Tukey's function is usually selected to be $t = 4.6851$ to achieve 95% asymptotic efficiency of the normal distribution.

Huber's function is smoother than Tukey's as shown in Fig. 2.6, which usually implies a faster convergence. However, Tukey's function provides a more aggressive treatment of outliers. A good strategy as recommend by Huber [177] is start the optimization with a smoother influence function and then perform a final pass with a more aggressive function such as Tukey's.

An in-depth treatment of robust statistics can be found in the book by Huber [177]. Theoretical considerations and history of the development of influence functions is provided

[11] Statistical efficiency is measure of estimators optimality. A more efficient an estimator is, the fewer measurements it needs.

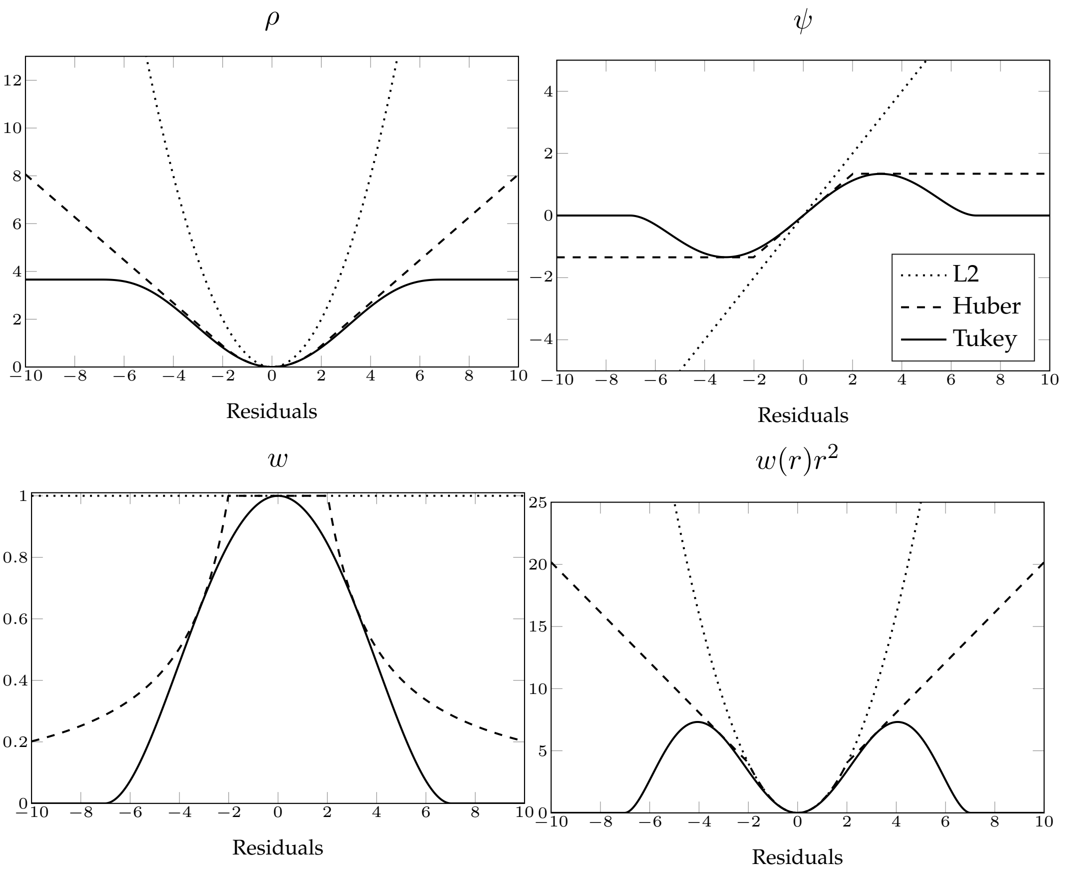


Figure 2.6: *Illustration of commonly used M-estimators in comparison to ordinary least-squares. The y-axis for least-squares has been truncated for a better visualization.*

by Hampel et al. [153]. A well-written summary covering the different robust estimation methods commonly used in Computer Vision is provided by Stewart [348]. Finally, an excellent tutorial on robust estimation with a thorough treatment of M-estimators for Computer Vision application is provided by Zhang [404], which also discusses how to robustly estimate the scale (standard deviation) of the residuals. When using robust M-estimators, the optimization procedure is commonly referred to as Iteratively Re-Weighted Least-Squares (IRLS). Some implementation details for IRLS optimization can be found in the works of Holland and Welsch [169] and Street et al. [350].

A practical example of nonlinear optimization is the original Lucas and Kanade algorithm (LK) for parametric image alignment [235], which is one of the most widely used algorithms in Computer Vision. LK and its variants are summarized in the next section.

2.5 Parametric Image Alignment

The goal of image alignment is to compute the deformations between one, or more, input images with respect to a fixed template/reference image such that a measure of dissimilarity between the template and the input images is minimized.

When the form of deformation between the template and input images is known, the problem is often referred to as *parametric* image alignment. Examples of applications of parametric image alignment include template tracking [248], corner localization [329], image registration, as well as direct Visual Odometry (VO).

In the sequel, we will provide a summary of the Lucas-Kanade (LK) algorithm [235] for parametric image alignment. It is important to note that the original LK formulation is an application of nonlinear least-squares. Some of LK's variants, however, perform "tricks" specific to the form of warps that do not lend themselves to standard nonlinear programming. For a complete exposition of LK and its variants we refer the reader to the excellent series of publications by Baker and Matthews [25] and Baker et al. [26, 27, 28, 29].

2.5.1 Original LK (Forward Additive)

Given two images, a template/reference $\mathbf{I}_0$ and an input/current image $\mathbf{I}_1$ related via a parametric transform/warp, we desire to estimate the parameters of the warp such that a function of intensity dissimilarity between the template and input is minimized.

The warp is a geometric transform that transfers pixel coordinate from the reference image $\mathbf{x}$ to the coordinate frame of the input image $\mathbf{x}'$. We will denote this warp with

$$\mathbf{w} : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n \tag{2.65}$$

$$\mathbf{x}' = \mathbf{w}\left(\mathbf{x}; \boldsymbol{\theta}\right), \tag{2.66}$$

where $\boldsymbol{\theta} \in \mathbb{R}^p$ is the vector of parameters we desire to estimate, $\mathbf{x} \in \mathbb{R}^m$ is a pixel coordinate in the frame of the template image, and $\mathbf{x}' \in \mathbb{R}^n$ is a pixel coordinate in the coordinate frame of the input image. Typically, pixel coordinates belong to the 2D image plane, *i.e.* $m = n = 2$, but other forms of warps can operate on higher dimensions depending on the problem formulation. A summary of commonly used warps is shown in Table 2.1.

Table 2.1: *Summary of common warps used in LK. For the projective warp (homography) we show the linear 8-parameter model, but other parameterizations are possible [30]. Jacobian for the rigid-body warp depends on the specific parameterization* (cf. *Eq.* (3.11))*. The function* $\pi\left(\cdot\right)$ *denotes the projection onto the image plane.*

	DOF $\boldsymbol{\theta}$	Input $\mathbf{x}$	Output $\mathbf{x}'$	Warp $\mathbf{w}(\mathbf{x};\boldsymbol{\theta})$	Jacobian $\frac{\partial \mathbf{w}}{\partial \boldsymbol{\theta}}$
Flow	$\begin{pmatrix} t_x \\ t_y \end{pmatrix} \in \mathbb{R}^2$	$\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$	$\begin{pmatrix} x' \\ y' \end{pmatrix} \in \mathbb{R}^2$	$\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{2\times 2}$
Affine	$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_6 \end{pmatrix} \in \mathbb{R}^6$	$\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$	$\begin{pmatrix} x' \\ y' \end{pmatrix} \in \mathbb{R}^2$	$\begin{pmatrix} 1+\theta_1 & \theta_3 & \theta_5 \\ \theta_2 & 1+\theta_4 & \theta_6 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$	$\begin{pmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{pmatrix} \in \mathbb{R}^{2\times 6}$
Homography	$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_8 \end{pmatrix} \in \mathbb{R}^8$	$\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$	$\begin{pmatrix} x' \\ y' \end{pmatrix} \in \mathbb{R}^2$	$\begin{pmatrix} 1+\theta_1 & \theta_2 & \theta_3 \\ \theta_4 & 1+\theta_5 & \theta_6 \\ \theta_7 & \theta_8 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$	$\begin{pmatrix} \alpha x & 0 & \alpha^2\gamma x & \alpha y & 0 & -\alpha^2\gamma y & \alpha & 0 \\ 0 & \alpha x & -\alpha^2\beta x & 0 & \alpha y & -\alpha^2\beta y & 0 & \alpha \end{pmatrix}$ $\alpha = (1+\theta_3 x+\theta_6)^{-1}\,,\ \beta = \theta_2 x + (1+\theta_5)y + \theta_8$ $\gamma = (1+\theta_1)x + \theta_4 y + \theta_7$
Rigid-body	$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_6 \end{pmatrix} \in \mathbb{R}^6$	$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \in \mathbb{R}^3$	$\begin{pmatrix} x' \\ y' \end{pmatrix} \in \mathbb{R}^2$	$\pi\left(\mathbf{T}(\boldsymbol{\theta})\mathbf{x}\right)$	$\frac{\partial \pi}{\partial \mathbf{x}'}\frac{\partial \mathbf{T}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{2\times 6}$

The goal of LK is to estimate the vector of parameters $\boldsymbol{\theta}$ such that

$$\mathbf{I}_0(\mathbf{x}) \equiv \mathbf{I}_1(\mathbf{x}'). \tag{2.67}$$

Exact equality, however, is unattainable due to noise and outliers. If we assume equality up to an additive Gaussian noise, then minimizing the sum of squared residuals corresponds to maximizing the likelihood, which is the optimal choice given no prior knowledge. Under least-squares, the minimization problem is of the form

$$\min_{\boldsymbol{\theta}} \sum_{\mathbf{x} \in \Omega_0} \|\mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{w}(\mathbf{x}; \boldsymbol{\theta}))\|_2^2, \tag{2.68}$$

where Ω_0 denotes a subset of pixel coordinates in the reference frame. This relationship is also known as the *brightness constancy assumption*, or brightness conservation [173].

Since the intensity value of a pixel, in general, is unrelated to the form of the warp, the optimization problem in Eq. (2.68) is nonlinear. Any nonlinear solver could be used to solve Eq. (2.68). In practice, Gauss-Newton (GN), or Levenberg-Marquardt (LM) are the algorithms of choice.

The solution proceeds by iteratively estimating a small parameter update $\Delta\boldsymbol{\theta}$ in the vicinity of a given initialization $\boldsymbol{\theta}$. A first-order Taylor series expansion about $\Delta\boldsymbol{\theta} = \mathbf{0}$ yields the following linear system of equations

$$\min_{\Delta\boldsymbol{\theta}} \sum_{\mathbf{x} \in \Omega_0} \|\mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}') - \frac{\partial \mathbf{I}_1}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta}\|_2^2. \tag{2.69}$$

Differentiating Eq. (2.69) with respect to $\Delta\boldsymbol{\theta}$ results in the expression

$$\sum_{\mathbf{x} \in \Omega_0} \left(\frac{\partial \mathbf{I}_1}{\partial \boldsymbol{\theta}}\right)^\top \left|\mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}') - \frac{\partial \mathbf{I}_1}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta}\right|. \tag{2.70}$$

The optimal solution for $\Delta\boldsymbol{\theta}$ is the critical point of the derivative (Eq. (2.70)) and obtained as the solution to the *normal equations* given by

$$\sum_{\mathbf{x} \in \Omega_0} \left(\frac{\partial \mathbf{I}_1}{\partial \boldsymbol{\theta}}\right)^\top \left(\frac{\partial \mathbf{I}_1}{\partial \boldsymbol{\theta}}\right) \Delta\boldsymbol{\theta} = - \sum_{\mathbf{x} \in \Omega_0} \left(\frac{\partial \mathbf{I}_1}{\partial \boldsymbol{\theta}}\right)^\top \big(\mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}')\big). \tag{2.71}$$

At the next iteration of the optimizer, parameters are updated additively:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}. \tag{2.72}$$

In the terminology of Baker and Matthews [25], the algorithm is called "Forward Additive." Forward, because the transformation maps pixel coordinates from the coordinate frame of the template to the coordinate frame of the input. Additive, because of the vector of parameters is updated additively after each iteration of the optimization algorithm.

The LK algorithm was invented early on in Computer Vision [235] for the purpose of estimating correspondences between the images of a stereo pair. Applications of LK now are numerous. The algorithm was also independently developed in Photogrammetry under the name: Least-Squares-Matching [144, 397]. An example implementation for the case of transitional shifts between the template and input images is shown in Algorithm 1.

Algorithm 1 The Lucas and Kanade algorithm

```
function t = lk(T, I)
  t = zeros(1, 2);
  [Ix, Iy] = gradient(I);

  it = 1;
  done = false;
  while ~done
    Ixw = imtranslate(Ix, t);
    Iyw = imtranslate(Iy, t);
    Ie  = T(:) - reshape(imtranslate(I, t), [], 1);
    J = [Ixw(:) Iyw(:)];
    dt = - J \ Ie;
    t = t + dt’;

    done = norm(dt) < 1e-6 || norm(J’*Ie, Inf) < 1e-8 || it > 50;
    it = it + 1;
  end

end
```

2.5.2 Inverse Compositional (IC)

The original LK algorithm is versatile and applicable to a variety of problems and warps. However, when the Jacobian of the warp is not constant, LK becomes computationally expensive. This is because the linearization step happens at the coordinate frame of the input image, which is warped (changes) at every iteration.

For a special set of warps, Baker and Matthews [25] devise an efficient algorithm by (conceptually) interchanging the roles of the template and input images. The warps must form a group to allow for a compositional update of the parameters. Instead of the original LK objective, Baker and Matthews [25] propose:

$$\underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathbf{x}\in\Omega_0} \|\mathbf{I}_0(\mathbf{w}(\mathbf{x};\Delta\boldsymbol{\theta})) - \mathbf{I}_1(\mathbf{w}(\mathbf{x};\boldsymbol{\theta}))\|^2, \tag{2.73}$$

with a parameter update performed using "inverse" composition:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \circ (\Delta\boldsymbol{\theta})^{-1}\,. \tag{2.74}$$

Inverting the estimated parameters after every iterations is necessary because the linearization is carried out at the coordinate frame of the template.

Performing a first-order expansion of Eq. (2.73), we obtain:

$$\sum_{\mathbf{x}\in\Omega_0} \|\mathbf{I}_0(\mathbf{w}(\mathbf{x};\mathbf{0})) - \mathbf{I}_1(\mathbf{w}(\mathbf{x};\boldsymbol{\theta})) + \frac{\partial \mathbf{I}_0}{\partial \boldsymbol{\theta}}\Delta\boldsymbol{\theta}\|^2 \tag{2.75}$$

$$= \sum_{\mathbf{x}\in\Omega_0} \|\mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}') + \frac{\partial \mathbf{I}_0}{\partial \boldsymbol{\theta}}\Delta\boldsymbol{\theta}\|^2, \tag{2.76}$$

where we assume that, without loss of generality, $\mathbf{w}(\mathbf{x};\mathbf{0})$ is the identity warp. The optimal update is obtained as the stationary point of the gradient and is given by the solution to the normal equations:

$$\sum_{\mathbf{x}\in\Omega_0} \left(\frac{\partial \mathbf{I}_0}{\partial \boldsymbol{\theta}}\right)^\top \left(\frac{\partial \mathbf{I}_0}{\partial \boldsymbol{\theta}}\right) \Delta\boldsymbol{\theta} = \sum_{\mathbf{x}\in\Omega_0} \left(\frac{\partial \mathbf{I}_0}{\partial \boldsymbol{\theta}}\right)^\top \big(\mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}')\big). \tag{2.77}$$

In IC, the Jacobian of the warp is evaluated at $\mathbf{x} = \mathbf{w}(\mathbf{x};\boldsymbol{\theta})$, with $\boldsymbol{\theta} = \mathbf{0}$:

$$\mathbf{J}(\mathbf{y};\boldsymbol{\theta}) = \left.\frac{\partial \mathbf{I}_0(\mathbf{y})}{\partial \boldsymbol{\theta}}\right|_{\substack{\mathbf{y}=\mathbf{x}_0\\ \boldsymbol{\theta}=\mathbf{0}}}. \tag{2.78}$$

The computational saving of the IC algorithm are significant. The Jacobian of the warp, and the inverse of the (Gauss-Newton approximation to the) Hessian need only be computed once at the beginning of the algorithm. The rest of the algorithm becomes a repeated application of image differences and matrix multiplication; operations that are efficient and amenable to parallelization.

The group requirement on the set of warps is not limiting. A large majority of warps commonly used in Computer Vision form a group, such as Perspective transformations (and their subgroups, *cf*. Table 2.1). Finally, IC, up to a first-order analysis, is equivalent to the original LK formulation [25].

Finally, we note that IC is not only more efficient, but its implementation is also simpler. However, given that the Jacobian is pre-computed and the warp is updated compositionally, it is not readily suited to commonly available nonlinear optimization packages (such as MINPACK [270] and others [48]). An implementation of the IC algorithm for translational warps is shown in Algorithm 2.

Algorithm 2 Baker and Matthews Inverse Compositional algorithm

```
function t = ic(T, I)
  t = zeros(1, 2);
  [Ix, Iy] = gradient(T);
  J = [Ix(:) Iy(:)];

  it = 1;
  done = false;
  while ~done
    Ie = T(:) - reshape(imtranslate(I,t), [], 1);
    dt = J \ Ie;
    t = t - dt';

    done = norm(dt) < 1e-6 || norm(J'*Ie, Inf) < 1e-8 || it > 50;
    it = it + 1;
  end

end
```

2.5.3 Other Variations

In addition to the IC algorithm, there are two more major variations. Firstly, is the forward compositional (FC) algorithm [332], where it is possible to pre-compute the geometric part of the Jacobian. Secondly, is the Efficient Second order Minimization (ESM) algorithm [40, 241]. The ESM algorithm obtains a second-order approximation of the Hessian efficiently by exploiting gradients from both the template and the input images. ESM has been claimed to have a wider basin of convergence [40], but it depends on the warp [110]. Mei et al. [253] develop further computational enhancements to the ESM method, and argue that ESM is more efficient than IC, even though each iteration of ESM is more computationally expensive. This is attributed to ESM's wider basin of convergence, hence requiring fewer iterations for convergence. A recent review of LK variations can be found in the work of Crivellaro et al. [84].

An important problem shared across all image alignment methods is the need to compute image gradients and interpolate the image intensities at subpixel locations, which we review briefly in the next section.

2.6 Image gradients and interpolation

The notion of image gradients is often taking for granted. It is however a rather involved topic. A digital image is often thought of as a discretized sampling of a continuous signal, but involves additional steps summarized in Fig. 2.7. Its gradient can thus be approximated using the method of finite-differences [79, ch. 7].

Forward, or backward finite-differences are insufficient when estimating the image gradient as they provide inconsistent results [124]. Due to the quantization step in the image acquisition pipeline, smoothing the image is often useful, which can be part of the derivative filter as well.

The gradient of a function is rotation-invariant. Farid and Simoncelli [117] make use of this rotation invariance requirement to develop constraints for optimal gradient estimation filter design and integrates smoothing as well. Another view on gradient estimation is based on the minimization of the quadratic energy functional as proposed by Weickert et al. [393].

We found the filters proposed by Farid and Simoncelli [117] to produce good results. A more efficient filter is the 5-point stencil used by Wedel et al. [391] for optical and scene flow estimation, and is given by $\frac{1}{12}\left[-1,\ 8,\ 0,\ -8,\ 1\right]$. It also produces good results. Notwithstanding, parametric image alignment problems studied in this work do not make nonlinear and nonlocal use of image coordinates, such as enforcing smoothing or other constraints. During our work, we found that in practice, a central difference operator to produce the best balance of accuracy and efficiency.

Interpolation is another important detail when working with subpixel positions. In general, projected image coordinates are real-valued and their exact value cannot be determined from a discrete image and hence must be interpolated. In fact, interpolation is

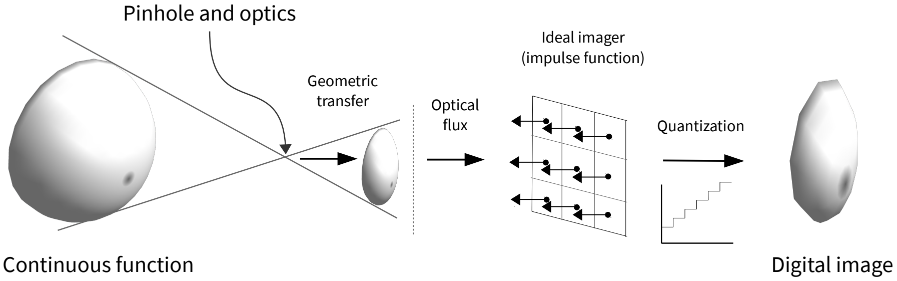


Figure 2.7: *Simplified view of the image acquisition step. Figure adapted from [187].*

so important that its use in the science can be traced back to the ancient Babylonian and Greek [254].

Thévenaz et al. [363] define interpolation concisely as the "recovery of continuous data from discrete data within a known range of abscissa." Interpolation differs from approximation by the requirement that an interpolated value at a known data point must match the value of known point.

Linear algorithms for interpolation represent the value of an interpolated function as a linear combination of known samples at integer locations as

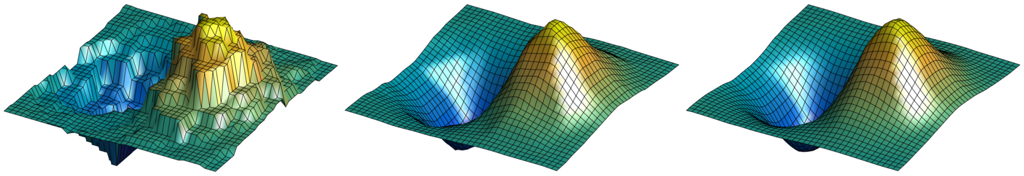
$$f(x) = \sum_{k \in \mathbb{Z}} f_k \phi\left(x - k\right), \tag{2.79}$$

where k the integer locations of the known samples denoted with f_k, $f(x)$ is the interpolated value of the function f at the desired location x, the function $\phi\left(\cdot\right)$ is the basis function, or the interpolation kernel.

For ϕ to be a valid interpolation kernel it must satisfy the interpolation constraint. Namely, when evaluated at integer location ϕ must be zero except at the origin where it takes one, *i.e.*, for $k \in \mathbb{Z}$,

$$f(k) = \begin{cases} 1 & \text{if } k = 0; \\ 0 & \text{otherwise.} \end{cases} \tag{2.80}$$

Popular interpolation functions include: nearest-neighbor, linear, and cubic. Nearest-neighbor



(a) *Nearest-neighbor* **(b)** *Linear* **(c)** *Cubic*

Figure 2.8: *Different interpolation kernels applied to the same grid.*

interpolation is a piecewise constant function (see Fig. 2.8a), its kernel is given by

$$\phi_{\text{nearest}}(x) = \begin{cases} 1 & \text{if } \frac{1}{2} \leq x < \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases} \tag{2.81}$$

Linear interpolation provides a smoother results (Fig. 2.8b) with the kernel

$$\phi_{\text{linear}}(x) = \big(1 - |x|\big)^{+}\,, \tag{2.82}$$

where $(\cdot)^{+}$ denotes taking the positive part only.

Finally, the cubic interpolation kernel, due to Keys [201], takes the form

$$\phi_{\text{cubic}}(x, a) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 < |x| \leq 1; \\ a|x|^3 - 5a|s|^2 + 8a|x| - 4a & 1 < |x| < 2; \\ 0 & 2 \leq |x|, \end{cases} \tag{2.83}$$

where a is usually -1, $-3/4$, or $-1/2$ depending on the smoothness requirements. Cubic interpolation results are depicted in Fig. 2.8c.

An excellent discussion on different kernels for image interpolation and their accuracy is provided by [378], where splines are shown to be a convenient tool capable of representing a range of smooth interpolation kernels. Implementation of common schemes is provided by Getreuer [140].

In the optical flow literature, interpolation artifacts have shown to affect accuracy. Cubic interpolation (either using Keys' formula Eq. (2.83), or using cubic B-splines) is a popular choice to reduce interpolation artifacts [352].

In the work presented in this document, we make use linear interpolation throughout for computational efficiency.

2.7 Visual Odometry and Vision-based SLAM

Visual odometry (VO) is the problem of estimating the pose of two, or more, cameras sharing a common field of view (FOV). Vision-based motion estimation is attractive because of the richness of visual information. A typical camera provide dense sampling of the scene, which in addition to pose estimation, can be used to carry out important tasks such as object recognition, and classification. In addition, cameras are relatively cheap, lightweight and require little power in comparison to other sensors such as lasers.

Vision-based SLAM (VSLAM) is a closely related problem. Some might even argue that both problems are the same. The trend, however, is to reserve the use of VO term to frame-frame motion estimation, while use the term VSLAM for a more complete system integrating multi-frame refinement (bundle adjustment), or loop closure.

Both, VO and VSLAM derive their name from robotics techniques for pose estimation [364]. Odometry is the process by which a robot estimate its position using sensory data. The term is associated with *wheel odometry*. The simplest case is perhaps a differential drive robot in the plane, which is equipped with two wheels. Given a known distance between the two wheels (the wheel base), the radius of each wheel, and the number of revolutions recorded from encoders, an estimate of the distance travelled can be obtained [49]. Integrating the odometry estimates over time provides us with a cumulative position of the robot. The process is known as *dead reckoning*.

Wheel odometry and dead reckoning are inaccurate due several factors such as wheel slippage. A spinning wheel does not always imply a moving robot. Wheels often slip, and in the extreme case, the robot may be stuck. To remedy this issue, exteroceptive sensing can be of great help. In particular, the use of cameras for position estimation and its demonstration on Mars [240].

Simultaneous Localization And Mapping (SLAM) is another fundamental problem in robotics [105, 219]. Given sensor measurements of an unknown environment, the SLAM problem is to use the measurements (and possibly control inputs) to simultaneously build a map of the environment and localize the robot to the map. Traditional sensors for SLAM problems in robotics include radar [102], sonar [362], and perhaps most accurate of all is lidar [12, 72, 268].

Using vision for SLAM is a natural progression. In Computer Vision, the problem is usually referred to a Structure-from-Motion (SFM) [38, 44, 300, 301], which predates the use of vision in SLAM, and can be traced back to early work in Photogrammetry. Central to

SFM accuracy in Computer Vision is the joint refinement of motion and structure in what is known as *bundle adjustment* (BA) [372]. The issue, however, was that BA is computationally demanding, and it was infeasible to perform for robotic applications demanding localization and mapping results in real-time.

Pioneering work in VSLAM [91] adapted commonly used and efficient filtering techniques in robotics to demonstrate the feasibility of real-time VSLAM. The map, however, consisted of only a few landmarks. This was later improved, also via filtering techniques, to a larger scale map [106]. The application of BA limited to two-views pose refinement was demonstrated by Nister et al. [288], where the term visual odometry was popularized.[12] Nister et al. [288] work is based on the feature-based pipeline [369]. Key to the real time application of the approach is the use of Single Instruction Multiple Data (SIMD) to quickly extract Harris corners [158].

Another key development in feature extraction is the introduction of the FAST keypoint detector [311], which demonstrates efficient keypoint extraction using a decision tree. FAST, was then used as the keypoint detector for an influential VSLAM system by Klein and Murray [205] and called PTAM. The key idea in PTAM (Parallel Tracking And Mapping) is to offload the task of accurate refinement using BA to a background thread. This means that real-time VO estimates can be obtained, while refinement is running simultaneously in a separate thread. The development of PTAM and other systems [111, 273] demonstrated the feasibility of BA in real-time systems. Advantages of BA in comparison to filtering was then established by Strasdat et al. [349], where it was concluded that BA is more efficient per unit of computation.

A summary of key developments over the past three decades is provided in Table 2.2. Fraundorfer and Scaramuzza [129], Scaramuzza and Fraundorfer [320] provide an additional discussion and review of different approaches to VO. To date, there are three mainstream approaches to VO and VSLAM. One, is the feature-based approach where the image is abstracted away in a few keypoints, or features [369]. Two, is the direct approach, where image data are used directly to estimate pose without the need for an intermediate representation [181]. Three, is the volumetric approach where range data are fused in 3D space [286]. The three methods are reviewed next.

[12] The term visual odometry appears in Biology, prior to its use in Vision and Robots, to describe the navigational abilities of honeybees [341].

Table 2.2: *Overview of key developments in VO and VSLAM. Other things we should have is image retrieval advances in robotics [85] and Computer Vision [338]. Biologically inspired systems for large scale topological SLAM [264]. Adaptation of bundle adjustment for online applications [189, 190]. BA is better than EKF [349]. The KITTI benchmark, which had a big influence on improving accuracy [138].*

	Beginnings
1979	Moravec's [269] pioneering work in visual navigation
1987	Stereo navigation [249]
1988	Horn and Weldon's [171] direct motion estimation
	Real-time
2002	Davison and Murray's [93] active vision approach
2003	Davison et al.'s [91], Davison's [92] mono SLAM
2004	Nister et al.'s [288] real-time mono VO, coins the term "visual odometry"
2005	Visual odometry success on Mars [240, 251]
2006	Klein and Murray [205] (PTAM) and Mouragnon et al. [273] show the feasibility of real-time bundle adjustment.
2007	Comport et al.'s [74, 75] direct/dense
	Dense
2010	Newcombe et al. [285] introduce a dense mapping system
2011	Izadi et al. [184], Newcombe et al. [286] adapt laser-based volumetric methods [86] to dense depth data from the Kinect
2012	Whelan et al. [395] introduce a moving volume to map larger spaces
	Quality Open Source Systems
2014	Engel et al. [110] present a semi-dense full VSLAM
2015	Mur-Artal et al. [276] present a feature-based full VSLAM
	Future
	Event based cameras [202, 274], beyond geometry and integrating semantics [71, 126, 305, 315, 386], and Chapter 7

2.7.1 Volumetric VO

Volumetric methods were developed for 3D surface reconstruction from laser data [86] where a signed distance function is used to fuse 3D measurements. The signed distance function is an implicit representation of the surface, where points off the surface are assigned a positive value, points inside the surface are assigned a negative value, while points at the surface are assigned a constant value, usually a zero. The surface can then be obtained as the level set of the signed distance function [296].

Newcombe et al. [286] presented the first real-time volumetric fusion system (KinectFusion) making use of high framerate depth measurements from the Kinect. The fused surface is continuously updated and refined by registering newly acquired data. The approach is commonly known as registration to a global model.

KinectFusion produces high quality reconstruction in small work areas [184, 286]. When the environment is small and static, the approach is virtually drift-free. This is because the world model can be maintained in memory. To map larger spaces, Whelan et al. [395, 396] introduced Kintinuous, a moving volume allowing larger space to be reconstructed densely.

Multiple real-time volumetric fusions systems exist currently with focus on object modeling from mobile devices in real-time [192, 295, 303, 360, 407]. While the volumetric approach can produce high quality reconstruction, it is in general limited to bounded environments. In particular, a bounding volume is needed to construct the signed distance function. Volumetric methods are also sensitive to the quality and accuracy of depth and normals [113].

2.7.2 Feature-based Visual Odometry

The feature-based pipeline to visual odometry has a well established research record dating to early work in Photogrammetry [146] for determining the pose of a calibrated camera from 3D–2D correspondences, or what is commonly known at the Perspective-N-Points (PnP) algorithm [156], or space resection [145].

The minimal solver for the PnP is obtained when the intrinsic parameters of the camera and the 3D coordinates of the world points are known using 3 points, or P3P [157]. Determining the pose using P3P and ransac [123][13] has been shown to be accurate for visual odometry

[13] While the seminal paper by Fischler and Bolles [123] is well known for the introduction of RANSAC in Computer Vision, its main motivation was a solution to the P3P problem.

from stereo [11, 288].

An efficient extension of the P3P algorithms to the general case of n points was proposed by Lepetit et al. [220], Moreno-Noguer et al. [271], which has found various applications in Computer Vision, especially for object pose determination [73].

Due to the importance of pose estimation from 3D–2D correspondences, the different solutions have been studied at length by [17, 120, 135, 157]. However, as the camera calibration and the coordinates of the scene points are not always available several extensions to the problem have been developed in the literature [141, 147, 212, 358, 406], with extensions to multi-cameras [208, 217, 385] and rolling shutter pose estimation [8, 317]. A collection of these methods is maintained by Pajdla and Kukelova on the web http://cmp.felk.cvut.cz/mini/. Implementations of commonly used algorithms is provided by Moulon et al. [272] and Kneip and Furgale [207].

Applications of feature-based methods for visual odometry are also well-established and reviewed by Fraundorfer and Scaramuzza [129], Scaramuzza and Fraundorfer [320]. The first application to visual odometry on a mobile robot was presented by Moravec [269], where the Moravec interest point operator was introduced [269, ch. 5]. A monocular algorithm to visual odometry was presented by Harris and Pike [159] and followed by the introduction of the Harris interest point detector [158], which remains popular to date.

2.7.3 Direct Visual Odometry

Recently, with the introduction of the Kinect [403], direct methods [181] have resurfaced to produce robust, (semi-)dense and real-time algorithms for Visual Odometry (VO) [21, 74, 75, 110, 128, 199, 232, 256, 257, 345, 374, 375]. At their core, direct VO algorithms are an application of the Lucas-Kanade [235] (LK) algorithm with a nonlinear warp. The nonlinearity of the warp is the result of the reliance on depth as well as the perspective projections required to obtain the image of a 3D point onto a rigidly moving camera. The LK algorithm, development, and variations are rich and versatile. In this review, we will focus on applications of LK to VO and VSLAM.

Amongst the first approaches to direct VO is the Quadrifocal warping algorithm by Comport et al. [74, 75]. The authors avoid the reliance on depth by exploiting the quadrifocal tensor between the four view of a rigidly moving stereo rig. Working with the quadrifocal tensor is complicated due to the high number of degrees of freedom. Comport et al. [75] rely on the quadrifocal tensor decomposition as two fundamental matrices and a trifocal tensor as their core warp. Recent work, however, has shown that non-linear warping with inaccurate depth can be used [21, 110, 199]. In fact, depth estimates need not be dense [10]. Even when the number of pixels used is sparse, the accumulated point cloud generates a densely populated 3D representation of the scene with enough fidelity for various robotic perception tasks as shown in Figs. 3.1, 6.25 and 6.26.

The Microsoft Kinect has been influential in the re-introduction of direct methods for VO. This is due to the availability of *dense* depth estimates along with RGB imagery in real-time. While using intensity-only constraints provide sufficiently accurate VO, one can also incorporate depth constraints. When using depth, however, one is presented with the challenge of having to compute a depth gradient, which may not be possible if depth estimates are sparse or compromised with large noise. A summary of the different constraints used in Direct VO is provided in Table 2.3.

Also, it is possible to use other sensors besides RGB-D in the versatile direct framework. A summary of different sensors used in direct VO is shown Table 2.4.

Table 2.3: *Summary of the different constraints used in Direct VO*

Constraint type	
Intensity only	Alismail and Browning [10], Audras et al. [21], Comport et al. [74, 75], Klose et al. [206], Lovegrove et al. [232], Meilland et al. [257], Newcombe et al. [285], Scandaroli et al. [319], Steinbrucker et al. [345]
Depth only	Bylow et al. [62], Canelhas et al. [64], Fang and Scherer [114], Fang et al. [116], Jaimez and Gonzalez-Jimenez [185]
Intensity and depth	Engel et al. [109, 110], Gutierrez-Gomez et al. [148], Kerl et al. [198, 199], Meilland et al. [256, 257], Tykkälä et al. [374, 375]

Table 2.4: *Summary of sensor type*

Sensor type	
RGB-D	Fang et al. [116], Klose et al. [206], Steinbrucker et al. [345]
Monocular	Daftry et al. [87], Engel et al. [109, 110], Lovegrove et al. [232], Newcombe et al. [285]
Stereo	Alismail and Browning [10], Comport et al. [74, 75], Engel et al. [107], Omari et al. [294]
Omnidirectional	Caruso et al. [66], Mei et al. [253], Meilland et al. [256]

This direct approach to VO has the following advantages in contrast to feature-based: (i) More robustness in degraded scenes, (ii) virtually parameter free, and (iii) the ability to produce richer 3D reconstruction of the scene without additional computational cost. However, limitations include (i) the need for small motion between frames, as the core part of the algorithm relies on linearizing the cost function, and (ii) consistent appearance between frames, or *brightness constancy*.

The first limitation, namely the need for small motions, is readily addressed with modern hardware. Modern cameras operate at frame rates in excess of 60fps which produces densely sampled video for most robotic tasks. Even when inter-frame motion is not sufficiently small, one can resort to scale-space [224], which improves the basin of convergence and provides a sound method to address large motions. However, the second limitation, brightness constancy, is more challenging. In fact, illumination change is a major obstacle in efforts to extend direct methods over multiple views with real data.

In the following section we present a summary of previous work that attempts to tackle the issue of multiple view direct optimization and the problem of violations of the brightness constancy assumption in an LK-based framework.

2.7.4 Direct Estimation of Structure & Motion from Multiple Views

The power of feature-based BA arises from the ability to refine the estimate of camera position and scene structure over multiple views [372]. In this work, we aim at attaining the same level of accuracy using image data without relying on keypoint processing. Our goal is to increase the robustness of VSLAM and allow vision-only algorithms to perform in challenging environments where keypoint extraction and accurate localization are not always possible.

Previous work on the estimation of the camera motion and scene structure over multiple views in a direct framework can be categorized into four main methods: (1) Alternating optimization of the state vector [], (2) Filtering framework [165, 166], (3) Simplifying assumptions [155, 213, 243, 291, 292], and (4) Reduction to keypoint-based geometric BA.

Alternating optimization

By alternating optimization we mean estimating a group of the desired variables separately while holding constant the rest. In the context of VSLAM, parameter groups include: rotation, translation and scene structure. The process is repeated for each of the parameters in turn until convergence. Examples of (direct) alternating minimization algorithms include the work of Mandelbaum et al. [243], Stein and Shashua [344], as well as Hanna [155], Oliensis [291, 292].

The use of alternating optimization is motivated by computational efficiency. When a depth estimate is required per pixel, simultaneous estimation of all variables becomes computationally prohibitive due to the sheer number of pixels in an image.

While an alternating framework can be shown to work in some scenarios, it is difficult to examine the optimality of the solution. Also, it is challenging to characterize the convergence properties of the algorithm. In visual structure-from-motion (SFM) tasks the optimal solution is attained by *joint* optimization of the state vector [372]. If accuracy of the algorithm is the most important, then slightly reducing the density of reconstruction in favor of an optimal solution is recommended. In fact, this joint optimization (when performed correctly) is more efficient than other heuristics [372]. Hence, we avoid alternating frameworks due to their limitations and focus on joint estimation algorithms.

Filtering/recursive estimation

Filtering is a common and useful technique in Robotics and Computer Vision. In filtering approaches, previous estimates are "marginalized out" and information is summarized with a probability distribution [349].

The filtering approach (also called recursive, or casual) to direct VSLAM has been previously adopted by various researchers including Heel [165, 166] as well as Barron and Eagleson [34]. The approach is also common in the context of multiple view stereo reconstruction, where the optimization objective is to estimate the depth per visible pixel assuming known camera motion [250, 382, 389].

Recent work by Strasdat et al. [349], however, have shown that BA is a more profitable strategy for VSLAM per unit of computation. In addition, optimality conditions in BA framework are easier to satisfy than a nonlinear filtering framework.

Simplifying assumptions

If one can assume the existence of certain structures in the scene, then it is possible to perform direct, joint multiple view VSLAM efficiently. The most common assumption is the existence of planar structures. The use of the planar world assumption has a rich history in Robotics and has been applied early on for robot navigation tasks [281]. Recent work by Silveira et al. [335] and Mei et al. [253] demonstrate methods using multiple views to improve direct VSLAM using planes. When the scene is composed of planes, the warp becomes linear and more efficient to implement. Furthermore, planar patches can cover a large surface in the world that typically projects onto a large area in the image. When the world is mostly planar, it is also possible to perform Bundle Adjustment over multiple views as shown by Kaess [188],Ataer-Cansizoglu et al. [19], Taguchi et al. [357] and Salas-Moreno et al. [316]. One can also include blur and fold-in image degradations terms as part of the optimization [252]. Subsequently, one can process a large number pixels in the image in one go.

Of course, this assumption breaks down when the world is not composed of planar segments. Addressing general scenes with the planarity assumption becomes a challenging problem.

Another common use of planes in direct SFM is planar-parallax. If the motion of a plane (or a selection of planes) in the world has been compensated for, then we can estimate the depth of non-planar points by their parallax to a reference plane. Irani et al. [182] have demonstrated a multi-view direct method that improves on the two-frame case [213, 318]. Nonetheless, planar-parallax is not suitable for VSLAM as it relies heavily on the existence of sufficiently large planar surfaces in the world. Furthermore, the main goal of planar-parallax algorithms is the estimation of 3D structure, which is only a sub-problem of VSLAM.

Another closely related approach to direct VSLAM is the Volumetric method [184, 286, 395, 396]. Volumetric methods have shown excellent results for real-time dense VSLAM, albeit requiring sophisticated GPUs. Volumetric methods fundamentally rely on imposing a working volume in 3D space. This limits their applications to indoor environments, or scenes with a known range of depth. For instance, it would be challenging to apply a purely volumetric approach to unstructured outdoor scenes where many of the strong rotation constraints are located on the plane at infinity. Meilland et al. [257] provide an excellent summary of the benefits and limitation of volumetric methods as well as an approach to integrate some advantages of volumetric methods alongside a direct framework.

Reduction to geometric BA

Finally, one can combine the benefits of direct methods with the power of geometric Bundle Adjustment as demonstrated by Forster et al. [128]. Direct VO can be used to obtain precise estimates of camera pose in real-time. In a parallel thread, a selection of geometric keypoints are tracked, and keyframed over time in order to be used later in a sliding window geometric BA. However, this approach does not provide a solution to VSLAM in challenging environments that lack distinctive keypoints. Hence, the drift reducing properties of multiple views can only be used in scenes were keypoint processing is possible.

2.7.5 Violations of the Brightness Constancy Assumption

Since the seminal work of Lucas and Kanade [235] and Horn and Schunck [173] various algorithms have been developed to address the limitations of the brightness constancy assumption, especially in optical flow estimation [352]. In this section, we briefly review methods in the literature that address violations of the brightness constancy assumption. Algorithms can be categorized into the following: (1) Not relying on the brightness constancy assumption by using illumination insensitive objective functions, (2) estimating illumination variations as part of the optimization problem, (3) eliminating illumination artifacts using an image preprocessing step (pre-filtering).

Illumination insensitive objective

Perhaps the most intuitive solution to the brightness constancy assumption is to use an illumination insensitive objective. Such objective functions include the normalized cross correlation (NCC), and the Mutual Information (MI) [307].

Mutual Information has been successfully applied to register images form different modalities in the Lucas-Kanade framework [90, 103] as well as tracking of known 3D models [81, 297]. Similarly, maximizing the correlation coefficient has been applied to multi-modal image alignment [180], template tracking [112, 319] as well as structure-from-motion [243].

While robust cost metrics with intrinsic independence on illumination can handle challenging scenarios, their main limitations are twofold. One, accuracy of results relies heavily on an accurate approximation of the Jacobian (and Hessian) of the cost function. In many instances, the closed-form derivation of the Hessian yields a numerically non-positive definite matrix, which is not suitable for optimization. The other limitation is that we can no longer rely on least-squares optimization. Instead, one must solve the problem with a general nonlinear optimization method. In template tracking problems, the number of variables is small and hence a general purpose optimizer provides satisfactory results. However, in the context of VSLAM the dimensionality of state-vector is large. In such scenarios, general purpose optimizers become slow and convergence is typically harder to attain when far from the optima.

More precisely, for large nonlinear optimization problems, we desire to formulate the problem as least-squares if possible. Least-squares problems are characterized by the *small residual* property. During linearization higher order derivatives of the objective vanish in comparison to the first order term ($\mathbf{J}^\top\mathbf{J}$). This implies a "free" and good estimate of the Hessian by only evaluating first-order partial derivatives. There exists a number of algorithms that take advantage of the special structure of least-squares problem to provide good time and convergence guarantees. In fact, solving least-squares problem can be considered a mature technology that we would like to exploit [289, ch. 10].

Estimating illumination variations

The most common approach to handle appearance variations is to model them explicitly as a multiplicative and an additive term. This is commonly known as gain + bias model [26, 36, 150]. In the context of VSLAM, the (global) gain + bias model can account for illumination changes arising from the camera exposure control. This has been shown to perform well in a frame-frame direct VO from Kinect data [206]. However, in order to address more complicated appearance variations, we need to estimate the gain per pixel, or per region in the image. Hence, the main disadvantage of the gain and bias model is the explosion of the dimensionality of the state vector due to the additional photometric parameters.

Consider, for example, a multiple view optimization problem with 5000 3D points seen by 5 views. If we represent each camera with 6-vector, and each 3D point by a 3-vector, then the state vector is in $\mathbb{R}^{15030\times 1}$. In order to make use of the gain and bias model, and without any prior assumptions on the scene, we may need to estimate a gain variable per 3D point per view and a bias term per image. This increases the state vector dimension by $5 \times 5000 + 5$ variables (assuming a gain variable per pixel and image) to become $\mathbb{R}^{400035\times 1}$. This larger problem is now more difficult and more computationally demanding to solve.

Alternatively, one could estimate the illumination of the scene given a surface model. However, this is a "chicken-and-egg" problem as surface reconstruction is not a priori available. Other methods can be used to estimate illumination without a surface reconstruction [35, 65, 104, 215, 377] but typically rely on certain assumptions on scene, the material types, or require RGB data. Such requirements restrict the utility of a general VSLAM algorithm.

Image pre-processing techniques

Pre-processing the input images to eliminate illumination artifacts is another alternative to illumination insensitive image alignment. For example, pre-normalizing the image to be of zero mean and unit standard deviation is equivalent to a global gain and bias estimation step. However, this form of normalization can only address changes to due to camera shutter control, and fails to address other complicated variations arising from shadows and specular reflections.

Other approaches augment the registration with higher order information from the image such as first- and second-order gradient information [61, 279, 336] (or other filters [236]). Additional higher-order information improve robustness against illumination artifacts, but require fine tuning and selecting appropriate balancing weights for the intensity terms versus the other terms. Moreover, extraction of higher order information from the image may exaggerate the sensor noise if the imagery is of low quality.

Another approach to illumination robust image alignment is using phase instead of intensity [125]. Phase is amplitude invariant. Hence, its robustness to changes in gain and bias. Nonetheless, spatial support near discontinues and occlusions is a limitation of phase-based methods [124].

Other techniques aim at applying simple filters to the image [80, 238]. However, they typically require RGB data and designed to work under natural sunlight illumination. They also "wash out" many of the useful frequencies in the image.

In optical flow estimation, Wedel et al. [390] propose the application of structure-texture decomposition to eliminate illumination artifacts from the image. The decomposition, however, is an iterative process that may not be suitable for online applications. Another risk of applying pre-filtering operations is that the filter might eliminate frequencies required to estimate certain degrees of freedom [180].

2.8 Geometric (Keypoint-based) Bundle Adjustment

Bundle adjustment (BA) [372] is a well-established and mature optimization framework. Triggs et al. [372] provides an excellent mathematical coverage of the topic. Dellaert [96], and Hartley and Zisserman [160] provide a tutorial aimed for implementers. Due to its popularity in Robotics and Computer Vision, several software packages are available [6, 230]. Variations on BA with emphasis on efficiency and handling large scale problems can be found in papers by Agarwal et al. [4], Kaess et al. [189, 190], Sibley et al. [333].

In the next section we present our completed work. The first part is the development of a direct VO algorithm for stereo data using disparity space warp function. The second is using (binary) feature descriptors to address illumination change in template tracking problems under arbitrary illumination variations.

In the next two chapter we present our approach to direct VO from stereo data (Chapter 3) and formulation of direct multiple view VSLAM (Chapter 7).

2.9 Vision in Challenging Environments

There is no shortage of challenging problems in robot vision. Many of the difficult challenges are due to illumination. For instance, something as seemingly insignificant to a human, such as shadows, can wreak havoc on many applications [216]. Removing shadows has been shown to improve the performance of image-based localization Corke et al. [80], but shadows are only a part of a bigger issues related to inconsistent illumination.

Maddern et al. [238] develop an illumination invariant representation from RGB images, with focus on eliminating shadows. The approach is demonstrated as a fall back option to improve the robustness of a visual odometry algorithm, when illumination artifacts hinder the feature-based pipeline. Nonetheless, research in visual odometry and vision-based SLAM in poorly lit environment has received little attention in the literature. Work by Milford et al. [262] proposes the use of low resolution images in combination with place recognition to improve vision-based SLAM in outdoor environments. According to the authors, robustness of the approach has been attributed to the use of image patches in lieu of keypoints [263]. Closely related to vision-based SLAM is optical flow and stereo, where Meister et al. [259] develop a benchmark dataset that closely resembles real world driving conditions. The dataset demonstrates the need for more robust correspondence estimation method sin vision as the current state-of-the-art remains sensitive to inconsistent inter-frame appearance.

It is often possible to pre-map an environment using a variety of sensors, such as a robot can the map for localization and place recognition purposes. Localization problems from vision-only data are arguably challenging as the mapped environments are constantly changing. In particular, the problem of localization across seasons has received a considerable attention in the literature [22, 33, 69, 280, 284, 353, 387].

Challenging situations in robot vision are very prevalent especially for dull, dirty and dangerous tasks [347]. Pose estimation in such environments has been traditionally performed with a suite of sensors. Recent work, however, has shown a good performance of pose estimation in smoke occluded environments [3]. Robustness to smoke was achieved by relying on depth measurements from an RGB-D sensor and "dehazing" the images [118] prior to feature extraction.

One of the goals of our work is to enable vision-only pose estimation in challenging data with minimal assumptions on the type of illumination as possible. Hence, the algorithms developed herein do not rely on pre-preprocessing as determining the appropriate filters is usually tightly coupled to the application domain.

Chapter 3

Direct Visual Odometry in Disparity Space

Contents

3.1 Summary of Contributions

- We develop a robust approach to visual odometry by using disparities directly.
- Numerically the approach is more stable than 3D points triangulated from stereo. This is particularly useful when working with distal observations.
- The approach is demonstrated on a range of datasets and is shown to be robust even when working with thumbnail-sized stereo.
- The implementation of the proposed method works faster than real-time.
- Essential to the speed of the runtime is a simple pixel selection procedure demonstrating that direct VO could perform well without the need for (semi-)dense data.

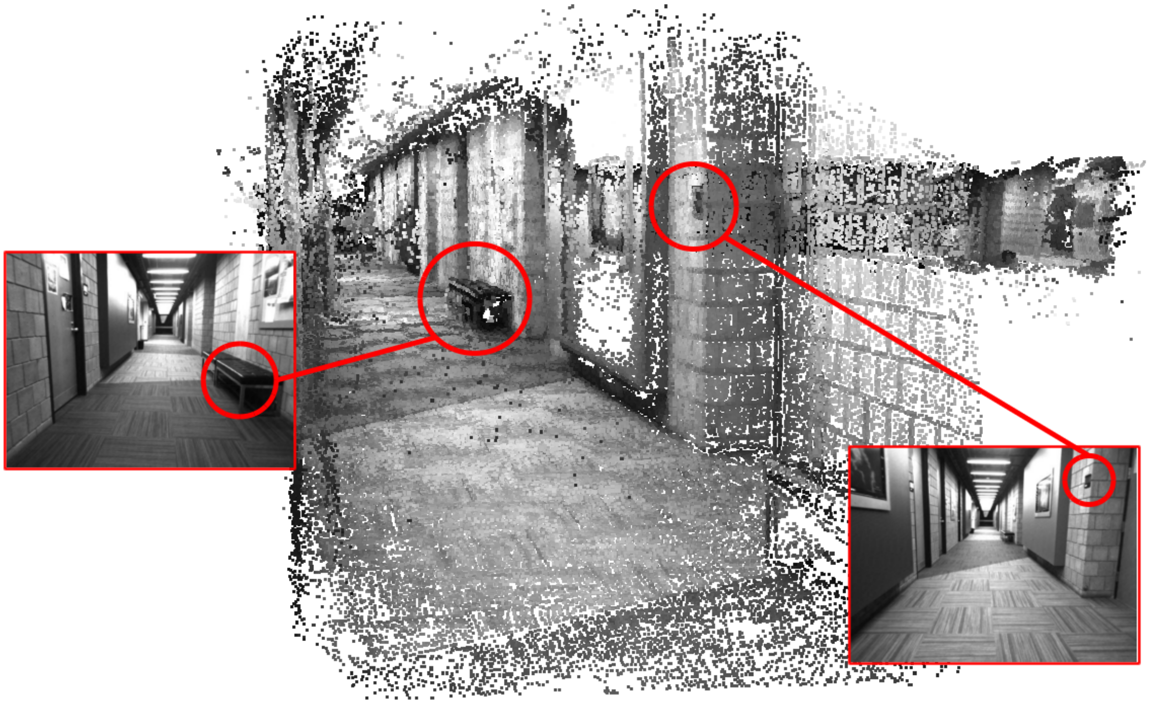


Figure 3.1: Example of dense 3D reconstruction of our indoor dataset.

In this chapter we provide a summary of our Direct Disparity Space (DDS) algorithm. Additional detailed analysis and experimental results are available in a separate publication [10].

3.2 Disparity space

Consider a rectified stereo image with baseline B and an upper triangular camera intrinsic matrix composed of the camera focal length f and the principle point $\mathbf{c} = (c_u, c_v)^\top$. Without loss of generality, let the left image be the origin of the coordinate system. A point $\mathbf{x} = (x, y, d, 1)^\top$ is an element of disparity space, where $x = u - c_u$, $y = v - c_v$ and $d = u - u_r$ is the disparity; the difference between the u-coordinate in the left image and its corresponding coordinate in the right image. Given this rectified stereo, the depth of an image point can be obtained with $Z = Bf/d$.

Consider two stereo pairs related via a rigid body transformation $\mathbf{T}(\boldsymbol{\theta}) \in SE(3)$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$, where p is typically 6, such that a 3D point $\mathbf{X} = (X, Y, Z, 1)^\top$ is transformed into $\mathbf{X}' = \mathbf{T}(\boldsymbol{\theta})\mathbf{X}$. This rigid-body motion relationship may be expressed in disparity space as

$$\mathbf{x}' \equiv \boldsymbol{\Gamma}\mathbf{T}(\boldsymbol{\theta})\boldsymbol{\Gamma}^{-1}\mathbf{x}, \tag{3.1}$$

where $\equiv$ denotes projective equality up-to-scale, and $\boldsymbol{\Gamma}$ is a 4×4 matrix that depends on the known stereo calibration and is given by

$$\boldsymbol{\Gamma} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & fB \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{3.2}$$

Demirdjian and Darrell [100] analyze the disparity space and show that it is a *projective space* $\subseteq \mathbb{P}^3$ with the important property that the measurement noise of the coordinates $\mathbf{x}$ is well-approximated with a Gaussian distribution.

3.3 Direct Visual Odometry

Let the intensity of a point $\mathbf{x}$ at the *reference* frame be given with $\mathbf{I}(\tilde{\mathbf{x}}) \in \mathbb{R}$, where $\tilde{\mathbf{x}} = (x + c_u = u, y + c_v = v)^\top$. With an abuse of notation, we will use $\mathbf{I}(\mathbf{x}) \coloneqq \mathbf{I}(\tilde{\mathbf{x}})$.

After a rigid-body motion with $\mathbf{T}(\boldsymbol{\theta})$, we obtain the *input* image $\mathbf{I}'(\mathbf{x}')$. Given an initialization $\boldsymbol{\theta}$, we seek to estimate a $\Delta\boldsymbol{\theta}$ — a small increment of pose parameters relating the two cameras — such that we minimize the sum of squared intensity error, or the *photometric error* given by

$$\Delta\boldsymbol{\theta}^* = \underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathbf{x}\in\boldsymbol{\Omega}} \left\| \mathbf{I}'\left(\mathbf{w}(\mathbf{x}; \boldsymbol{\theta} + \Delta\boldsymbol{\theta})\right) - \mathbf{I}\left(\mathbf{x}\right) \right\|^2, \tag{3.3}$$

where Ω is a subset of pixel coordinates of interest in the reference frame, and $\mathbf{w}(\cdot)$ is a *warping* function that depends on the parameter vector we seek to estimate. After every iteration, the current estimate of parameters is updated via an additive rule (*i.e.* $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$). This process repeats until convergence, or some termination criteria have been met.

This formulation is the standard Lucas-Kanade (forward additive) algorithm [235] (see

Section 2.5.1). An efficient variation on the Lucas-Kanade algorithm is Baker & Matthews' Inverse Compositional (IC) algorithm (see Section 2.5.2). The IC algorithm makes two modifications to the error function that significantly improve efficiency. First, is to interchange the roles of $\mathbf{I}$ (the reference/template image) with $\mathbf{I}'$ (the input/current image). The other, is to compound incremental estimates using a compositional update rule instead of an additive one. Under the IC formulation we seek an update of the parameters $\Delta\boldsymbol{\theta}$ such that

$$\Delta\boldsymbol{\theta}^{*} = \underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathbf{x}\in\mathbf{I}} \|\mathbf{I}\left(\mathbf{w}(\mathbf{x};\Delta\boldsymbol{\theta})\right) - \mathbf{I}'\left(\mathbf{w}(\mathbf{x};\boldsymbol{\theta})\right)\|^{2}. \tag{3.4}$$

The optimization problem in Eq. (3.4) is nonlinear irrespective of the form of the warping function or the parameters. To obtain a solution, we perform a first-order Taylor expansion and arrive at the following closed form (normal equations):

$$\Delta\boldsymbol{\theta} = \left(\mathbf{J}^{\top}\mathbf{J}\right)^{-1}\mathbf{J}^{\top}\mathbf{e}, \tag{3.5}$$

where $\mathbf{J} = \left(\mathbf{g}(\mathbf{x}_1)^{\top}, \ldots, \mathbf{g}(\mathbf{x}_m)^{\top}\right) \in \mathbb{R}^{m\times p}$. Here, m is the number of pixels and $p = |\boldsymbol{\theta}|$ is the number of parameters. Each $\mathbf{g}$ is $\in \mathbb{R}^{1\times p}$ and is given by

$$\mathbf{g}(\mathbf{x}) = \nabla\mathbf{I}(\mathbf{x})\frac{\partial\mathbf{w}}{\partial\boldsymbol{\theta}}, \tag{3.6}$$

where $\nabla\mathbf{I} = (I_u, I_v) \in \mathbb{R}^{1\times 2}$ is the image gradient along the u- and v-directions. Finally,

$$\mathbf{e}(\mathbf{x}) = \mathbf{I}'(\mathbf{w}(\mathbf{x};\boldsymbol{\theta})) - \mathbf{I}(\mathbf{w}(\mathbf{x};\Delta\boldsymbol{\theta})) \tag{3.7}$$

is the *error image*.

At the next iteration of the optimization algorithm, parameters of the motion model are updated via the IC rule given by

$$\mathbf{w}\left(\mathbf{x},\boldsymbol{\theta}\right) \leftarrow \mathbf{w}\left(\mathbf{x},\boldsymbol{\theta}\right) \circ \mathbf{w}\left(\mathbf{x},\Delta\boldsymbol{\theta}\right)^{-1}. \tag{3.8}$$

We refer the reader to the excellent series by Baker & Matthews [25, 26] for a detailed treatment.

3.3.1 Algorithm

Given a reference image $\mathbf{I}$ with an associated disparity map and an input image after camera motion $\mathbf{I}'$, we seek to estimate the parameters of motion such that the expression in Eq. (3.4) is minimized. The warping function is given by:

$$\mathbf{w} : \left(\mathbb{P}^3 \times \mathbb{R}^6\right) \to \mathbb{R}^2 \tag{3.9}$$

$$\mathbf{w}(\mathbf{x}, \boldsymbol{\theta}) = \pi\left(\boldsymbol{\Gamma}\mathbf{T}(\boldsymbol{\theta})\boldsymbol{\Gamma}^{-1}\mathbf{x}\right) + \begin{pmatrix} \mathbf{c} \\ 0 \end{pmatrix}, \tag{3.10}$$

where $\pi\left(\cdot\right)$ performs homogeneous division to bring back the point to Euclidean space. Finally, we add back the principle point $\mathbf{c}$ to obtain 2D pixel coordinates in the image plane.

In order to use a direct approach, we need to compute an analytic expression of the Jacobian with respect to the parameters around the identity $\boldsymbol{\theta} = \mathbf{0}$. Using the Lie algebra parameterization of rigid transformations, *i.e.* a *twist*, $\boldsymbol{\theta} = \left(\omega_x, \omega_y, \omega_z, \nu_x, \nu_y, \nu_z\right)^\top \in \mathbb{R}^6$, we obtain the Jacobian of the warping function in Eq. (3.10) per point $\mathbf{x}$ as [170]:

$$\nabla\mathbf{I}\frac{\partial \mathbf{w}}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\mathbf{0}} = \mathbf{g}(\mathbf{x}) = \begin{pmatrix} -fI_y + \alpha y/f \\ fI_x - \alpha x/f \\ yI_x - xI_y \\ \beta I_x \\ \beta I_y \\ \alpha\beta/f \end{pmatrix}^\top \in \mathbb{R}^{1\times 6}, \tag{3.11}$$

where $\nabla\mathbf{I} = \left(I_x, I_y\right)$ is the image gradient, $x = c_u - u$, $y = c_v - v$, $d = u - u_r$, with

$$\alpha = xI_x + yI_y, \quad \text{and} \quad \beta = d/B. \tag{3.12}$$

For m pixels, we stack the values of Eq. (3.11) into an $m \times 6$ matrix and obtain an update of parameters $\Delta\boldsymbol{\theta}$ by solving the normal equations in Eq. (3.5). After every iteration the pose estimate is updated using the inverse compositional update rule given by Eq. (3.8).

3.3.2 Robustness

The least-squares optimization (Eq. (3.5)) is sensitive to outliers. In order to obtain a robust estimate we replace the squared error with a *robust* cost function. Choice of the robust function is rather arbitrary and can only be determined experimentally [404].

We experimented with several cost functions and found Tukey's bi-weight [39] to perform the best. This is possibly due to suppressing high residuals instead of only reducing their influence. The bi-weight function for a residual $r_i \in \mathbb{R}$ and parameter/cutoff threshold $\tau \in \mathbb{R}$ is given by

$$\rho(r_i; \tau) = \begin{cases} \left(1 - \left(r_i/\tau\right)^2\right)^2 & \text{if } |r_i| \leq \tau; \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$

The cutoff threshold τ is set to 4.6851 to obtain a 95% asymptotic efficiency of the normal distribution. The threshold assumes normalized residuals with unit deviations. For this purpose, we use a robust estimator of standard deviation. For m observations and p parameters, the robust standard deviation is given by:

$$\hat{\sigma} = 1.4826 \left[1 + 5/(m - p)\right] \operatorname*{median}_i |r_i|. \tag{3.14}$$

The constant 1.4826 is used to obtain the same efficiency of least-squares under Gaussian noise, while $\left[1 + 5/(m - p)\right]$ is used to compensate for small data [404]. In practice, $m \gg p$ and the small data constant vanishes.

In summary, given a list of residuals $\mathbf{r} = (r_1, \dots, r_m)^\top$, where each residual is given by:

$$r_i = \mathbf{I}'\left(\mathbf{w}(\mathbf{x}_i; \boldsymbol{\theta})\right) - \mathbf{I}\left(\mathbf{w}(\mathbf{x}_i; \Delta\boldsymbol{\theta})\right), \tag{3.15}$$

we compute a robust estimate of the standard deviation $\hat{\sigma}_{\mathbf{r}}$ using Eq. (3.14) and compute the weight per residual as $w_i = \rho\left(r_i/\hat{\sigma}_{\mathbf{r}}\right)$. By concatenating the weights into an $m \times m$ diagonal matrix $\mathbf{W}$, we may obtain an estimate of the parameters at every iteration by solving with the following *weighted* normal equations:

$$\left(\mathbf{J}^\top \mathbf{W} \mathbf{J}\right) \Delta\boldsymbol{\theta} = \mathbf{J}^\top \mathbf{W} \mathbf{e}. \tag{3.16}$$

3.3.3 Pixel selection

Traditionally, direct methods are associated with the concept of dense, or semi-dense, algorithms that make use of all possible pixel information. Intuitively, the use of as many as possible data points could increase robustness. However, the large number pixels typically used in direct algorithms incur a high computational cost necessitating implementation on parallel architectures such as high-end GPUs.

The literature on pixel selection for direct pose tracking is sparse. In the case of optical flow, the seminal work of Shi & Tomasi [329] introduced a feature selection method based on the "textureness" of the patch surrounding the pixel. The textureness score is obtained by analyzing the Eigenvalues of the design matrix, which is composed of image gradients.

For pose tracking applications, Dellaert & Collins [97] propose a method that selects pixels that constrain each degree-of-freedom (DOF) the most. A known motion prior is required, however. Meilland *et al*. [256] propose an alternative based on recursively sorting each dimension of the Jacobians and greedily keeping elements with the highest magnitude. Simpler methods include discarding pixels with a gradient magnitude smaller than a fixed threshold [206].

In this work, we show that direct camera tracking *need not* be dense. By reducing the number of pixels, the algorithm runs in real-time on a single CPU core without compromising accuracy or robustness. Our pixel selection is based on the feature "binning/bucketing" idea common to feature-based methods [288], where the image is virtually split into a grid/buckets, and a certain number of pixels with strong *cornerness* score is kept in each bucket.

In our case, the influence of a pixel correlates, to an extent, with its gradient magnitude. For example, a pixel with no gradient does not contribute to the optimization as its contribution to the Jacobian in Eq. (3.11) vanishes. Hence, we perform our pixel selection using the gradient magnitude as a substitute for the cornerness map. Pixels with a local gradient magnitude maxima in a neighborhood of 3×3 pixels are used for pose estimation. In contrast to feature-based methods, we do not enforce a maximum number of features per grid cell.

This choice of pixel selection (and others) will be evaluated thoroughly in Chapter 4, where we show that the non-maxima suppression strategy is beneficial for estimating the rotation of the camera.

3.3.4 Additional implementation details

Our algorithm does not require elaborate parameter tuning or specialized heuristics. The only tunable parameters are the stereo algorithm parameters, which depend on the dataset. We use a basic block matching stereo (as implemented in OpenCV[1]). Stereo parameters include, SAD window size and disparity range. To address large motions (and speed up the convergence rate) the algorithm is implemented in a scale space pyramid. We do not scale down the disparity image to avoid interpolation across occlusion boundaries. Instead, disparities for coarser levels of the pyramid are interpolated from the disparity map computed at the finest level using nearest-neighbor interpolation. Each level of the pyramid is scaled by a factor of $1/2$ of the previous image size and smoothed with a Gaussian filter prior to downsampling with bilinear interpolation. Convergence is determined if the norm of the estimated parameters is less than 1×10^{-6}, change in parameters is less than 1×10^{-8} or a maximum number of iterations is reached. The maximum number of iterations was set to 300 on the finest level, and 50 for all other levels.

Image gradients are computed using the usual central differences. There are different methods to compute image gradient, which we will evaluate in Chapter 4.

3.4 Experiments and Results

In this section, we evaluate the performance of our algorithm on different datasets including outdoor and indoor environments. For the outdoor datasets we process all frames. That is, we do not perform any keyframing, even when the robot is stationary. We also do not perform any global optimization/bundle adjustment or make use of other sensors. In the following, we will call our algorithm DDS: *Direct Disparity Space*.

3.4.1 KITTI data

We evaluate the performance of our algorithm on the KITTI odometry benchmark [138] in comparison to two open source algorithms targeted for robotic applications: (1) VISO2 [137] and (2) FOVIS [176]. We use both algorithms with the authors' default parameters, which perform well.

Results on KITTI data are summarized in Fig. 3.2. Our algorithm's average translation

[1] Using MATLAB R2013b http://www.mathworks.com/help/vision/ref/disparity.html

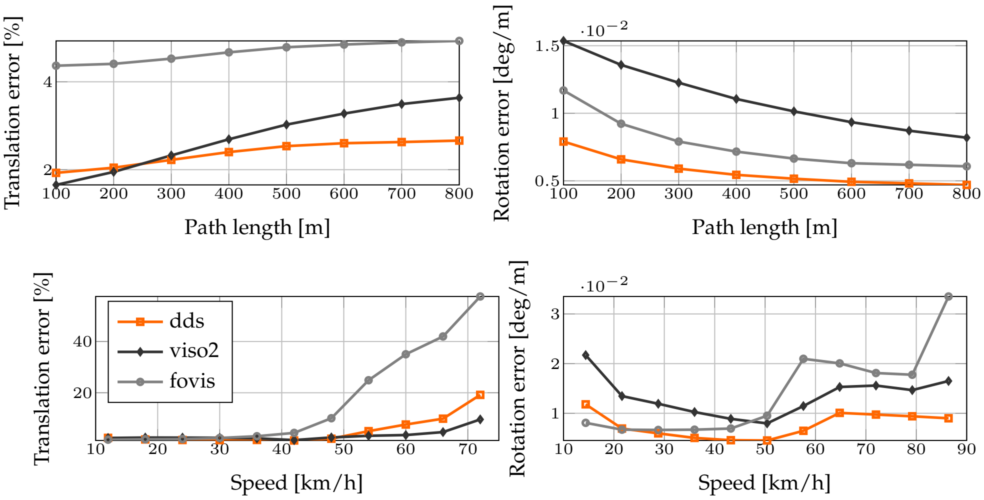


Figure 3.2: *Results for the* 11 *training sequences of the KITTI odometry benchmark.*

error is 2.35% and the average rotation error is $0.0058°/m$, which are accurate for a frame–frame method without pose initialization. In particular, our rotation error is close to, and sometimes better, than some multi-frame methods on the KITTI benchmark. The main sources of error appear in estimating the translation of the camera at high vehicle speeds. High speed driving produces larger baseline between images and violates the small motion assumption.

Interestingly, rotation error for both FOVIS and ours (DDS) are better than VISO2. This is potentially due to more accurate rotation estimation results when using image intensities directly. In fact, most of VISO2's rotation drift appears to be in roll estimates and consequently camera height. We hypothesize that this is related to the small vertical FOV of the camera. In contrast, direct methods are able to better exploit this reduced FOV by not relying on the accuracy subpixel feature localization.

3.4.2 Wean hall (indoor data)

This dataset was collected with a Bumblebee2 stereo color camera of resolution $640 \times 480\,\text{px}^2$ at $\approx 30\,\text{Hz}$ [11] and is available online `http://www.cs.cmu.edu/~halismai/wean`. A summary of the data is shown in Table 3.1. The camera was mounted on a LAGR robot.[2] For ground truth, we use a 2D estimate of robot pose using calibrated wheel odom-

[2] See `http://www.nrec.ri.cmu.edu/projects/lagr/`. Our robot is slightly modified to use a single stereo camera titled towards the ground, and equipped with an accurate fiber optic gyro from KVH, model # DSP-3000.

Table 3.1: *Wean hall dataset summary*

focal length	$\approx 3.88\,\mathrm{mm}$	baseline	$0.12\,\mathrm{m}$
# frames	6510	distance	$\approx 294\,\mathrm{m}$

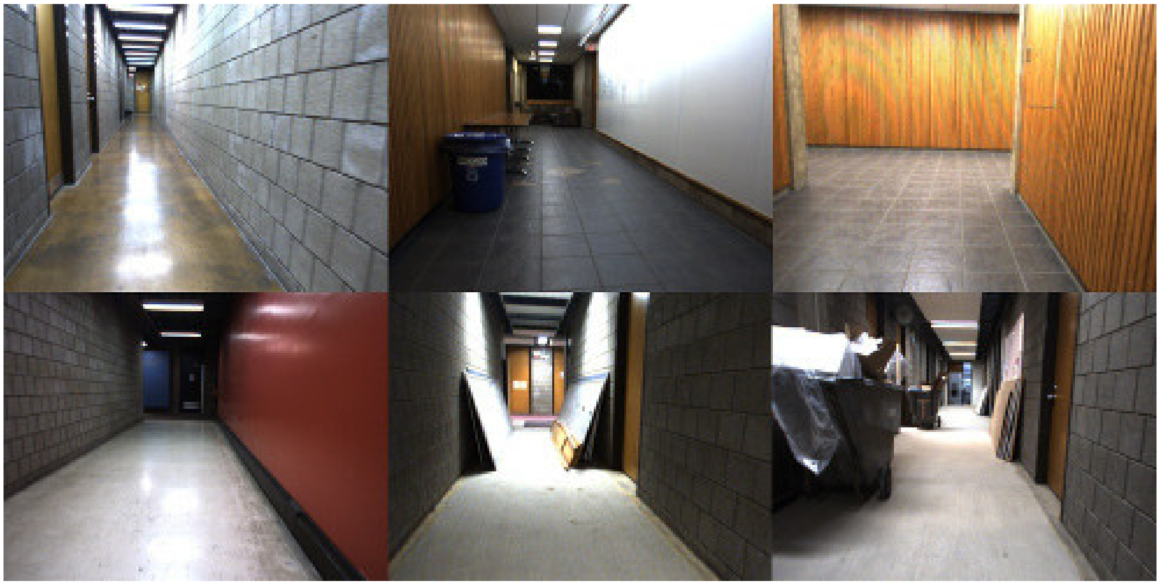


Figure 3.3: *Example images from the Wean hall dataset.*

etry combined with an accurate gyroscope. This is an approximate ground truth, but it is reasonable as the indoor environment is flat. The camera's raw output is a Bayer pattern, which we interpolate to a color image prior to use for motion estimation. The Bayer pattern causes a reduction in resolution in comparison to native grayscale output.

The dataset features strong specular reflections on the ground, lack of texture in some areas as well as high frequency repetitive texture in others. An example is shown in Fig. 3.3. The robot was driven at an average speed of $\approx 0.7\,\mathrm{m\,s^{-1}}$.

Due to the high framerate of the camera, we implement a keyframing strategy based on the magnitude of the estimated motion. The pose of each non-keyframe is initialized with the current estimate of pose until the motion magnitude is large enough. Upon keyframing, we reset the pose initialization to the identity. For the results shown here, we keyframe when the estimated translation magnitude is $30\,\mathrm{cm}$, or when any of the estimated rotation angles exceeds 5°. Results are shown in Fig. 3.3 and Fig. 3.4.

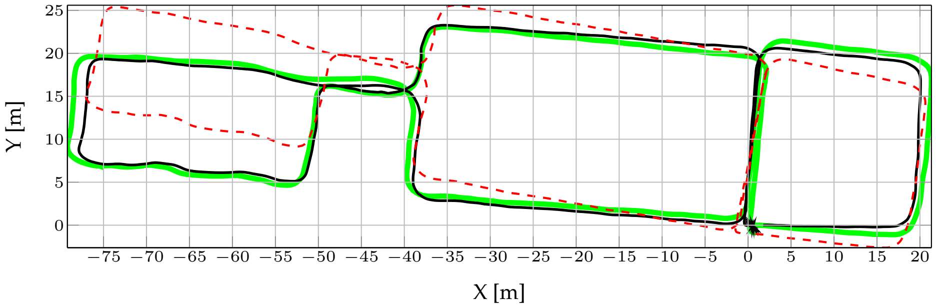


Figure 3.4: *Top view of estimated path for ground truth (🟩), VISO2 (- - -) and DDS (——).*

3.4.3 Dense 3D reconstruction

We demonstrate a simple scheme to obtain sufficiently dense 3D reconstruction using our algorithm. The output of our algorithm after every keyframe consists of an estimate of the camera pose, as well as the set of disparity space point and their IRLS weights upon convergence.

After every keyframe, we select points with Tukey weights from the third quartile (75 percentile) and with range of at most 30 m. We triangulate the points, and project them to the world coordinates using the current estimate of the keyframe's pose. As the pixel selection scheme is not based on features, the selected points over multiple frames do not correspond to a single 3D point space. Hence, the overlap between frames will consist of mostly distinct 3D points that produce a dense reconstruction of the environment.

Examples of our reconstruction are shown in Fig. 3.1, and Fig. 3.5. Note, disparity maps were obtained via block matching and include a large amount of noise and outliers. The 3D reconstruction results indicate the accuracy of the method over a short sequence of frames as well as robustness against outliers.

3.4.4 Visual odometry from stereo thumbnails

We also evaluate the robustness of the algorithm using low resolution images (178×54) and compare it against VISO2 using full resolution images (1241×376). Results are shown in Fig. 3.6. Even with the low stereo resolution our algorithm out performs VISO2 and remains accurate and robust.

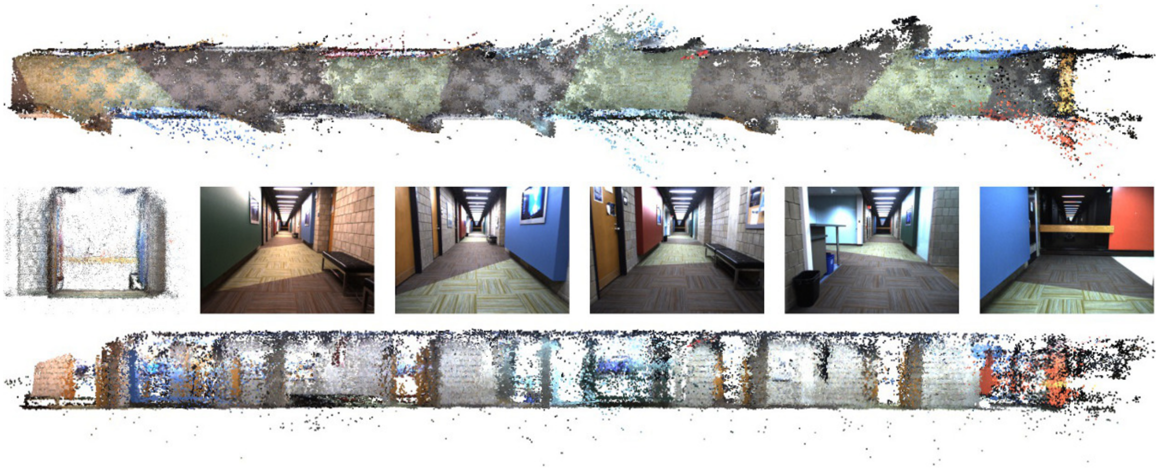


Figure 3.5: First row shows a top view from our indoor dataset. Images along the corridor are shown in the second row. The first image shows that floor of the model is flat as expected. The last row shows a side view of the corridor. The 3D points are obtained automatically from the algorithm without post-processing or filtering. The corridor length is $\approx 40\,\mathrm{m}$.

3.4.5 Pixel selection

Not all pixels contribute equally to the cost function; only a few pixels contribute towards the error function [97]. The simplest approach to pixel selection is to discard pixels with gradient magnitude less than a pre-specified fixed threshold. The rational is that a pixel with zero gradient does not contribute to the error function. However, the mere magnitude of the gradient is not a sufficient predictor of a pixel's performance. For instance, restricting the optimization to pixels with a high gradient magnitude might bias the solution in undesirable ways.

To illustrate this, we run our algorithm using all available pixels with an absolute gradient magnitude greater than a threshold. The average performance on the KITTI benchmark training data is illustrated in Fig. 3.7.

As shown in the figure, including all possible pixels is suboptimal. Similarly, selecting pixels with a very high gradient magnitude is suboptimal as well. A good threshold for the tested optics and the benchmark environment is within 10% of the image dynamic range.

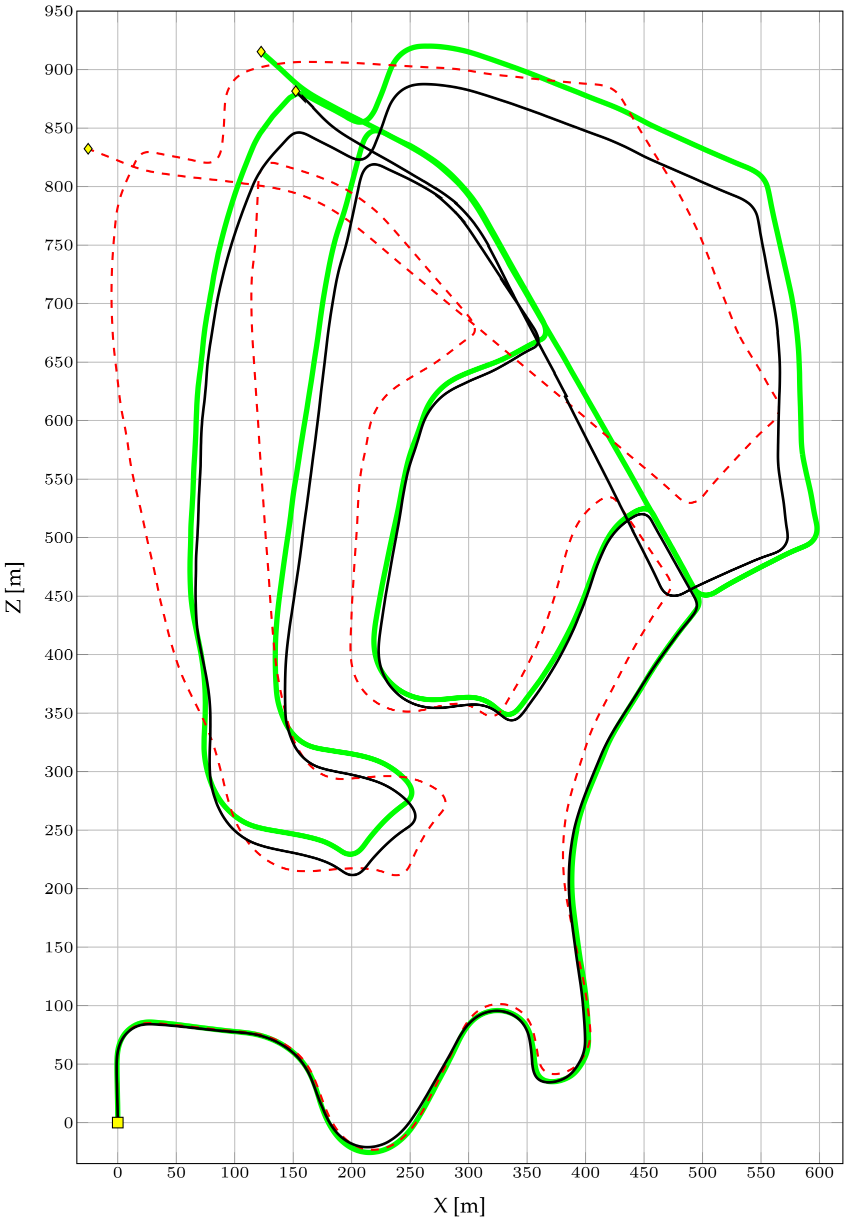


Figure 3.6: *Result from* KITTI *Seq. 02. Ground truth is shown in (▬), VISO2 path is in (- - -), and DDS is in (——). The (□) indicates the start of the sequence, and (◇) indicates the final location. Our results are generated from an image of size* 178 × 54*, while VISO2 results are generated from the full resolution* 1241 × 376*.*

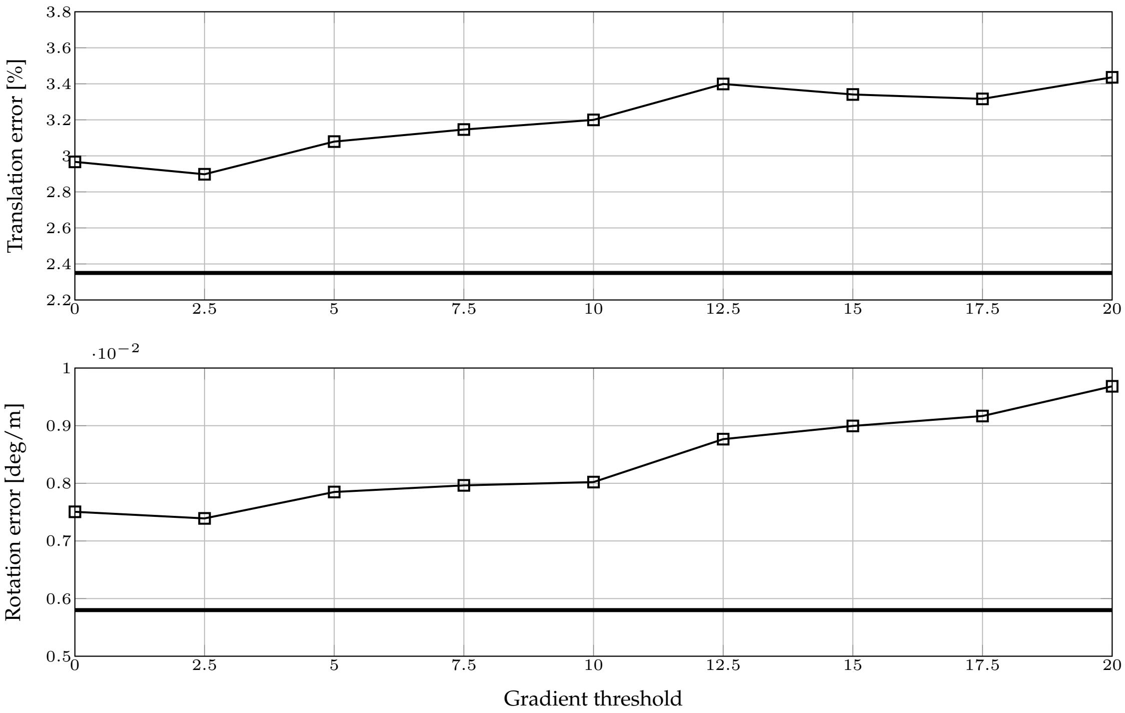


Figure 3.7: *Average performance on KITTI training data with different absolute gradient magnitude cutoff thresholds. The input images are converted to floating point prior to computing the gradients, and their range is kept* $\in [0, 255]$*. Detailed evaluation plots are shown in Fig. 3.8*

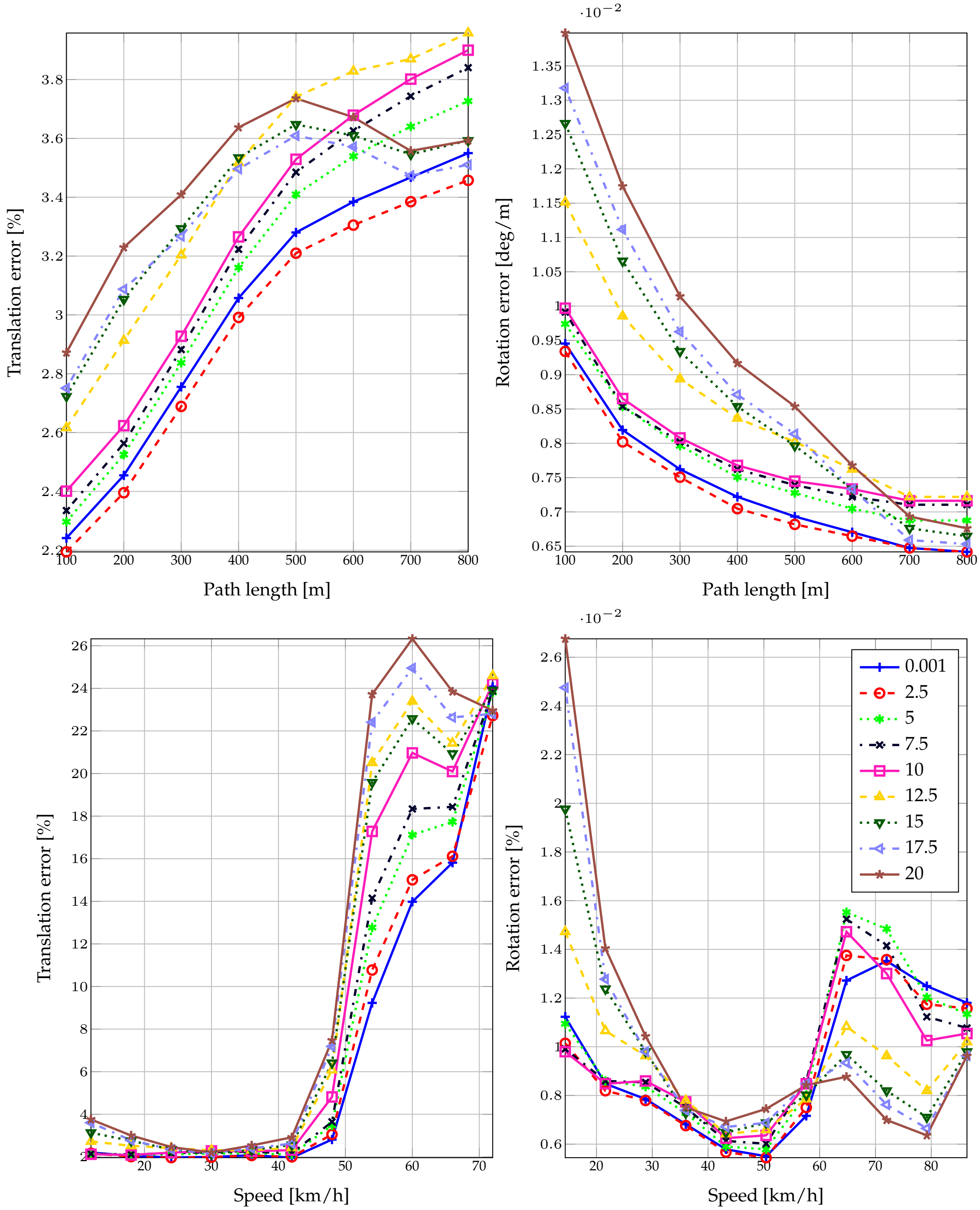


Figure 3.8: *Detailed performance on KITTI benchmark for various absolute gradient magnitude cutoff thresholds. See Fig. 3.7 for a summary.*

3.5 Discussion & Summary

Scene points on the plane at infinity are independent of camera translation and can be used for rotation estimation and calibration. In contrast to other work, our algorithm can make use of points at infinity without special handling. This is particularly useful for outdoor applications and we believe leads to improved rotation estimates.

We did not observe a need to use sophisticated stereo matching algorithms. Indeed, our stereo matching is very straightforward SAD block matching with limited disparity range resolution. Enhanced stereo may improve the accuracy and/or convergence speed at the expense of more computation time for stereo. Improvements are not guaranteed as (semi) global stereo methods may over smooth the estimated disparities. This issues remains to be experimentally validated.

In this work we dealt with the problem of pose estimation only (camera tracking). Two important improvements are possible. The first would be structure/disparity refinement. We can include disparity refinement in the same pose tracking framework by using observations from the right image. Another possibility is modeling disparities with some surface representation (*cf*. [335]). The second important improvement is integrating information from multiple frames in a bundle adjustment/filtering framework. This, in fact, is necessary to reduce drift over long sequences as we will show in Chapter 7.

In this chapter, we presented a *direct* framework for visual odometry using a warping function in *disparity space* (DDS). The algorithm is shown to be efficient, robust and accurate even with low resolution images. Experiments illustrate the applicability of the algorithm to various environments with little to no manual parameter tuning. Finally, we have also shown that direct camera tracking can achieve accurate and robust performance while using only a fraction of the image data via a simple pixel selection strategy.

In the next chapter we study the effect of the different implementation details on the accuracy of direct visual odometry. In particular, we study the effect of the image gradient estimation scheme, smoothness of interpolation for image warping, and the effect of pixel selection on accuracy.

CHAPTER 4

Evaluation of Direct Visual Odometry Details

Contents

4.1 Summary of Contributions

- We evaluate important details at the core of direct visual odometry algorithms. Namely, we evaluate the effect of interpolation, the numerical scheme used for the estimation of image gradients as well as the density of pixels (measurements) contributing to the optimization.

- The evaluation is carried out on a number of synthetic and real datasets.

- Surprisingly, there is little to no effect of the quality of interpolation on results.

- The effect of pixel selection and gradient estimation is more pronounced.

- Interestingly, when using Iteratively Re-weighted Least-Squares (IRLS) optimization, the minima of the photometric error does not coincide with maximizing the accuracy of parameters even in the absence of photometric variations.

- Based on the analysis presented in this Chapter, it is possible to devise more accurate direct visual odometry algorithms if the application domain is known beforehand.

4.2 Introduction

Visual odometry (VO) is the problem of determining the relative motion between two rigidly moving cameras sharing a common field-of-view. Due to its importance and wide array of applications, VO has received much attention in the literature [320]. The classical pipeline to VO is commonly referred to as the feature-based approach [276, 288, 369], where sparse interest points are matched between frames and used to estimate the camera motion in a robust estimation framework [123, 348].

Recently, however, with the increasing availability of high framerate cameras, direct methods for motion estimation [171, 181] have been shown as a viable alternative due to their robustness and speed [75, 107, 110, 198]. Unlike feature-based algorithms, direct methods can use much of the image to estimate a few degrees-of-freedom. Hence, they have been also called dense [199, 255], or semi-dense in the literature [109] depending on the number of pixels contributing to pose estimation.

Most direct VO algorithms are an adaption of the Lucas-Kanade (LK) algorithm [235] to nonlinear warps. The LK algorithm and its variants [25] aim to establish an approximate linear relationship between appearance and geometric displacements. The relationship between appearance and geometric displacement is seldom linear, so the linearization process is typically repeated until convergence.

Central to LK, and its variants, are two steps: *(i) linearization:* where the image gradient is stochastically estimated over the image lattice to achieve a first-order linearization about the current estimate of parameters; *(ii) warping:* where a suitable interpolation scheme is employed to evaluate image intensities at subpixel positions to compute the vector of residuals at the current iteration.

The two steps are important to accuracy and rate of convergence. In the optical flow literature [124, 173], for instance, it is well-known that bicubic interpolation is more accurate than its linear counterpart [61, 352]. Similarly, a direct link between the accuracy of gradient estimation and the accuracy of the estimated parameters has been established in other image-based optimization algorithms [117, 393, 393].

Yet, virtually all current LK-based direct VO algorithms rely on *central differences* to estimate the gradient of the image and *bilinear* interpolation for warping. These two choices are usually motivated by their computational efficiency (and ease of implementation). However, are they accurate enough? Or, are we missing additional accuracy by not exploiting more sophisticated schemes?

4.2.1 Contributions

In this work, we evaluate the different implementation details at the core of direct VO, which are often overlooked in favor of computational efficiency. In particular, we focus on the effect of the gradient estimation scheme as well as the smoothness of the interpolation kernel used for warping.

Additionally, we evaluate the effect of pixels sub-sampling, or pixel selection, on the accuracy of the system. Unlike dense motion estimation tasks, such as optical and scene flow, displacement vectors for all pixels are not requires as the goal of direct VO is to estimate the motion of the camera. Hence, we may focus the computational effort on a subset of pixels to achieve two gaols: (i) computational efficiency, and (ii) enhanced accuracy. This hypothesize has been partially validated for parametric image alignment tasks [59, 97] and feature tracking [329], but has received little attention in the context of direct VO [256].

4.3 Related Work

Estimating motion directly from images, without the need for an intermediate representation such as interest points, can be tracked back to the seminal works of Lucas and Kanade [235] and Horn and Schuck [173]. Its application to determining parametric image alignment tasks had been demonstrated for various applications [41, 181]. The direct approach is attractive due to its enhance precision, which is attributed to using numerous measurements to estimate a few degrees-of-freedom [181]. This additional redundancy is also useful to enhancing robustness in degraded imaging conditions such as motion blur.

A fundamental assumption in most direct methods is the need for small pixel displacements for the linearization step to be valid. Although the framerate of consumer cameras is ever increasing, motion induced displacements in the image are seldom fractional. Hence, the approach is usually implemented in scale-space [224] to enhance robustness and widen the basin of convergence when a suitable initialization is far from the local minima.

Recently, however, the use of direct VO for accurate and robust motion estimation has been on the rise with application to stereo [75, 107], mono [110], omni-directional [66], and RGB-D sensors [199, 346]. Applications of direct VO are not only limited to vision-only pose estimation, but has also have been demonstrated in visual-inertial navigation [47, 294] as well as rolling shutter rectification [200, 258]. Since the work we present here evaluates the core implementation details of direct VO, we expect that conclusions could be transfered to other application domains.

Optimization details of the LK algorithm and its variants have been studied in the literature, where it was shown that the Gauss-Newton, and the Levenberg-Marquardt algorithms, outperform the Netwon method for image alignment [25]. Due to image noise and outliers, robust estimation frameworks are essential for most applications [26, 43].

Estimating the image gradient is a crucial step in direct VO. To data, however, the dominant gradient estimation methods are: central-differences (first-order accurate 3–point stencil), the Sobel, and the Scharr operators. Gradient estimation has received more attention for optical flow estimation. For instance, it well-known that merely using forward-or, backward- differences is insufficient for accurate results [173], as the asymmetry provides inconstant results [117]. Central-difference filters are usually performed on Gaussian-smoothed images to eliminate some image noise. Nonetheless, the effect of this smoothing on the accuracy of the estimation has not been established. This issue will be examined in the experimental portion of this work.

There exists several recent research papers evaluating various aspects of direct VO, or direct camera tracking as it is sometimes called [285]. Fang and Scherer [115] evaluate several algorithms for motion estimation from RGB-D data in the context of autonomous micro air vehicles (MAV) navigation. The work demonstrates that direct methods are favorable when appearance variations are minimal.

The work by Klose *et al*. [206] studies the performance of direct tracking using RGB-D imagery as well. The work focuses on the different variations of the LK algorithm [25] and the effect of the M-estimator [404] on the precision and robustness of the system. We will make use of Klose *et al*.'s results [206] to guide the experiments conducted in this chapter.

Handa *et al*. [154] study the effect of the camera framerate on direct tracking performance. Under synthetically ideal conditions (infinite SNR), the work demonstrates increased accuracy as a function of framerate and image resolution. However, under realistic scenarios, increasing the framerate reduces the exposure time, which results in low quality imagery. Hence, as expected, there is a trade-off between framerate and accuracy.

Zia *et al*. [407] evaluate dense and semi-dense tracking from a system-level perspective. The work studies issues such as energy consumption as well as the choice of the computing platform (mobile versus desktop) and their effect on accuracy.

In the next section we provide a summary of the direct VO pipeline and a brief exposition of the different implementation details evaluated in this work.

4.4 Direct Visual Odometry Pipeline

Let the intensity of a pixel coordinate $\mathbf{p} = (u,\ v)^\top$ in the *reference* image be given by $\mathbf{I}(\mathbf{p}) \in \mathbb{R}$. After camera motion, a new image is obtained $\mathbf{I}'(\mathbf{p}')$. The goal of direct tracking is to estimate the camera motion parameters $\Delta\boldsymbol{\theta} \in \mathbb{R}^d$ such that the photometric error is minimized

$$\Delta\boldsymbol{\theta}^* = \underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathbf{p}\in\Omega} \left\| \mathbf{I}'\left(\mathbf{w}(\mathbf{p};\boldsymbol{\theta} \boxplus \Delta\boldsymbol{\theta})\right) - \mathbf{I}\left(\mathbf{p}\right) \right\|^2, \tag{4.1}$$

where Ω is a subset of pixel coordinates of interest in the reference frame, $\mathbf{w}\left(\cdot\right)$ is a *warping* function that depends on the parameter vector we seek to estimate, and $\boldsymbol{\theta}$ is an initial estimate of the motion parameters. After every iteration, the current estimate of parameters is updated additively (*i.e.* $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \boxplus \Delta\boldsymbol{\theta}$), where $\boxplus$ generalizes the addition operator over the optimization manifold. The process is repeated until convergence, or some termination criteria have been met [25, 235]. The warping function for direct VO used in this work is given by:

$$\mathbf{w}(\mathbf{p};\boldsymbol{\theta}) = \pi\left(\mathbf{T}(\boldsymbol{\theta})\mathbf{P}(\mathbf{p};d)\right), \tag{4.2}$$

where $\mathbf{P}(\cdot;\cdot)$ denotes the back-projected 3D point at pixel location $\mathbf{p} = (u, v, 1)^\top$ with an estimated depth value d. This point can be obtained as:

$$\mathbf{P}(\mathbf{p}, d) = d\left(\mathbf{K}^{-1}\mathbf{p}\right) \ \in \mathbb{R}^3, \tag{4.3}$$

$$\text{with } \mathbf{K} = \begin{pmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{pmatrix} . \tag{4.4}$$

Finally, the expression $\mathbf{T}(\boldsymbol{\theta})$ denotes a rigid-body transformation matrix. In this work, similar to the literature [198, 206, 346] we parameterize the rigid-body pose using the $\mathfrak{se}(3)$ Lie Algebra such that for $\boldsymbol{\theta} \in \mathbb{R}^6$, the pose can be obtained using the exponential map as:

$$\mathbf{T}(\boldsymbol{\theta}) = \exp(\sum_{j=1}^{6} \boldsymbol{\theta}_j \mathbf{G}_j) = \exp(\widehat{\boldsymbol{\theta}}) \ \in SE(3), \tag{4.5}$$

where $\mathbf{G}_i \in \mathbb{R}^{4\times4}$ is the i^{th} generator of $SE(3)$ Lie Group. In general, the exponential map must be approximated, however for $\mathfrak{se}(3)$, it a has a closed-form [277].

4.4.1 Linearization Algorithm

According to the terminology of Baker and Matthews [25], the optimization problem in Eq. (4.1) is the Forward Additive form of LK. Since computational efficiency is important for VO, the *Inverse Compositional (IC)* formulation [25] is often used. The IC objective takes the form

$$\Delta\boldsymbol{\theta}^* = \underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathbf{p}\in\Omega} \left\| \mathbf{I}'\left(\mathbf{w}(\mathbf{p};\boldsymbol{\theta})\right) - \mathbf{I}\left(\mathbf{w}(\mathbf{p};\Delta\boldsymbol{\theta})\right) \right\|^2 . \tag{4.6}$$

Under the IC formulation, the parameter update is given by $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \circ \Delta\boldsymbol{\theta}^{-1}$, where $\circ$ denotes composition. When using $\mathfrak{se}(3)$ to represent the motion of the camera, the update rule for the $k+1$ iteration takes the form

$$\boldsymbol{\theta}_{k+1} = \log\left(\exp(\boldsymbol{\theta}_k)\exp(-\Delta\boldsymbol{\theta})\right) \ \in \mathfrak{se}(3). \tag{4.7}$$

Image intensities are, in general, unrelated to their pixel coordinates. Hence, Eq. (4.6) is nonlinear irrespective of the warp. A solution proceeds by linearizing the objective using a 1^{st}-order expansion about the current estimate and computing the update as the critical point of the derivative of the linearized form. The derivative of the 1^{st}-order Taylor expansion of Eq. (4.6) takes the form

$$\sum_{\mathbf{p}\in\Omega} \frac{\partial \mathbf{I}}{\partial \boldsymbol{\theta}}^{\top} \left| \mathbf{I}'\left(\mathbf{w}(\mathbf{p};\boldsymbol{\theta})\right) - \mathbf{I}\left(\mathbf{w}(\mathbf{p};\mathbf{0})\right) - \frac{\partial \mathbf{I}}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} \right| . \tag{4.8}$$

Letting $\mathbf{J} \in \mathbb{R}^{n\times 6}$ be the Jacobian of the objective, obtained by stacking the partial derivatives of the image with respect to the warp from n observations, *i.e.*

$$\mathbf{J} = \left(\frac{\partial \mathbf{I}(\mathbf{p}_1)}{\partial \boldsymbol{\theta}}^{\top}, \cdots, \frac{\partial \mathbf{I}(\mathbf{p}_n)}{\partial \boldsymbol{\theta}}^{\top} \right). \tag{4.9}$$

And letting $\mathbf{e} \in \mathbb{R}^{n\times 1}$ be the vector of residuals at the current iteration given by $\mathbf{I}'(\mathbf{w}(\mathbf{p};\boldsymbol{\theta})) - \mathbf{I}(\mathbf{p})$, then the parameter update can be obtained by solving the normal equations

$$\mathbf{J}^{\top}\mathbf{J}\Delta\boldsymbol{\theta} = -\mathbf{J}^{\top}\mathbf{e}. \tag{4.10}$$

Equation (4.8) reveals the vast computational savings when using the IC formulation. Since the partial derivatives of the image with respect to the parameters ($\partial\mathbf{I}/\partial\boldsymbol{\theta}$) is evaluated at the (fixed) template, it needs computed only once.

Other variations on the original LK objective in Eq. (7.8) include the Forward Compositional [150], which allows part of the Jacobian to be pre-computed and the Efficient Second-order Minimization (ESM) [40], where a 2nd-order approximation to Hessian is obtained by averaging image gradients from both the template and input images.

The effect of LK variations in direct VO has been previously evaluated in the literature [110, 206], where it was shown that IC performs equally well and often better than other variants. Hence, we will use the IC formulation as the linearization method in our framework.

4.4.2 Image Gradients

Using the chain-rule, we may write the partial derivatives of the image with respect to the parameters as:

$$\frac{\partial \mathbf{I}}{\partial \boldsymbol{\theta}} = \nabla \mathbf{I} \frac{\partial \mathbf{w}}{\partial \boldsymbol{\theta}}, \tag{4.11}$$

where $\partial\mathbf{w}/\partial\boldsymbol{\theta} \in \mathbb{R}^{2\times 6}$ is the Jacobian of the warp. When using the IC formulation, the Jacobian must be computed at $\boldsymbol{\theta} = \mathbf{0}$, and $\mathbf{P} = (x, y, z)^{\top}$, which takes the following closed-form:

$$\frac{\partial \mathbf{w}}{\partial \boldsymbol{\theta}} = \frac{\partial \pi}{\partial \mathbf{P}} \frac{\partial \mathbf{T}(\boldsymbol{\theta})\mathbf{P}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{2\times 6} \tag{4.12}$$

$$= \frac{1}{z} \begin{pmatrix} f_u & 0 & -\frac{x f_u}{z} \\ 0 & f_v & -\frac{y f_v}{z} \end{pmatrix} \begin{pmatrix} [\mathbf{P}]_\times & \mathbf{I}_{3\times 3} \end{pmatrix} \tag{4.13}$$

$$= \frac{1}{z} \begin{pmatrix} f_u \frac{xy}{z} & -f_u \frac{x^2+z^2}{z} & f_u y & f_u & 0 & -f_u \frac{x}{z} \\ f_v \frac{y^2+z^2}{z} & -f_v \frac{xy}{z} & -f_v x & 0 & f_v & -f_v \frac{y}{z} \end{pmatrix} \tag{4.14}$$

where $[\mathbf{P}]_\times$ denotes the 3×3 skew-symmetric matrix obtained from the elements of $\mathbf{P}$. Here, we assumed that the first three elements of $\boldsymbol{\theta}$ correspond to the rotational part.

Letting expression $\nabla \mathbf{I} = (I_u, I_v)$ be the image gradient along the u- and v-axis respectively, we obtain the Jacobian of the objective in Eq. (4.11) as

$$\frac{\partial \mathbf{I}}{\partial \boldsymbol{\theta}} = \nabla \mathbf{I} \frac{\partial \mathbf{w}}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{F_v y^2 + F_u xy + F_v z^2}{z^2} \\ \frac{-(F_u x^2 + F_v xy + F_u z^2)}{z^2} \\ \frac{-(F_v x - F_u y)}{z} \\ F_u / z \\ F_v / z \\ \frac{-(F_u x + F_v y)}{z^2} \end{pmatrix} \in \mathbb{R}^{6\times 1}, \tag{4.15}$$

where $F_u = I_u f_u$, and $F_v = I_v f_v$.

In general, the image gradient cannot be obtained deterministically in closed-form, and must be estimated. The accuracy of gradient estimation is important to many image-based optimization problems [117]. In this work, we evaluate the following choices:

- **Central-differences (CD).** CD are derived from the definition of the derivative [79]. When applied to images, they can be computed as a convolution with the separable kernel $\frac{1}{2}[1, 0, -1]$. The kernel is often referred to as the 3-point stencil.
- **Central-differences with smoothing (CD-s).** To attenuate the effect of image noise, the image is often smoothed with a Gaussian prior to estimating the derivative. We use a 3×3 Gaussian kernel with $\sigma = 1$.
- **Sobel (SB**.) A popular method to estimate image gradients proposed by Sobel and

Feldman [340]. The Sobel kernel to estimate the derivative along the x-axis of the image is given by

$$\frac{1}{8}\begin{pmatrix}1 & 0 & -1\\ 4 & 0 & -4\\ 1 & 0 & -1\end{pmatrix}. \tag{4.16}$$

The filter is separable into $(\frac{1}{2}\left[-1,0,1\right]) * (\frac{1}{4}\left[1,2,1\right])$. The operator was originally intended for edge detection. Its popularity in gradient estimation possibly stems from the availability of high performance implementation in most image processing libraries.

- **Scharr (SH).** Proposed by Scharr [321] to produce more accurate gradient estimates from images. In this work, we use the 3×3 kernel given by

$$\frac{1}{32}\begin{pmatrix}3 & 0 & -3\\ 10 & 0 & -10\\ 3 & 0 & -3\end{pmatrix}. \tag{4.17}$$

The kernel is separable and can be obtained by a convolution with $\frac{1}{2}\left[-1,0,1\right] * \frac{1}{16}\left[3,10,3\right]$.

- **Five-point Stencil (S5).** This is the 5-point stencil version of the CD filter, which is more accurate as demonstrated in the optical flow literature [352, 390]. The filter is given by:

$$\frac{1}{18}\left[-1,8,0,-8,1\right]. \tag{4.18}$$

In numerical analysis, the filter coefficients are determined to minimize the approximation error of the derivative in terms of the Taylor series expansion. However, we found that making the coefficient sum up to one results in faster convergence as first-order gradient-based optimization is sensitive to scaling.

- **Farid & Simoncelli 5-tap optimal (FS-5).** Proposed by Farid & Simoncelli [117] such that the kernel preserves the rotational invariance of the gradient operator. The approach relies on two 5-tap kernels, one is for interpolation while the other is for differentiation.

- **Farid & Simoncelli 7-tap optimal (FS-7).** This is the 7-tap version, which achieves higher approximation accuracy of the gradient.

4.4.3 Interpolation for Image Warping

At every iteration of direct VO, we must evaluate the input image at sub-pixel coordinates to compute the *error image* with respect to the template. In this work, we consider "linear" methods of interpolation [363] due to there efficiency. Linear algorithms for interpolation represent the value of an interpolated function as a linear combination of known abscissas as

$$f(x) = \sum_{k \in \mathbb{Z}} f_k \phi\left(x - k\right), \tag{4.19}$$

where f_k is the known value of the function at the k^{th} integer location, $f(x)$ is the interpolated value of the function at the desired location x, and $\phi\left(\cdot\right)$ is the basis function.

For ϕ to be a valid interpolation kernel it must satisfy the interpolation constraint. Namely, when evaluated at integer location, it must be zero except at the origin where it takes unity, *i.e.*, for $k \in \mathbb{Z}$,

$$f(k) = \begin{cases} 1 & \text{if } k = 0; \\ 0 & \text{otherwise.} \end{cases} \tag{4.20}$$

In this work, we evaluate the effect of the following kernels:

- **Linear**, where the interpolation kernel is given by

  $$\phi_{\text{linear}}(x) = \big(1 - |x|\big)^{+}, \tag{4.21}$$

  where $(\cdot)^{+}$ denotes taking the positive part only. This is the most commonly employed interpolation method in direct VO due to its efficiency as it requires only two samples per dimension. Linear interpolation is continuous, but not differentiable. Hence, its C^0-continuous.

- **Cubic** interpolation, where the kernel is given by [201]

$$\phi_{\text{cubic}}(x, \alpha) = \begin{cases} (\alpha+2)|x|^3 - (\alpha+3)|x|^2 + 1 & 0 < |x| \leq 1; \\ a|x|^3 - 5\alpha|s|^2 + 8\alpha|x| - 4\alpha & 1 < |x| < 2; \\ 0 & 2 \leq |x|. \end{cases} \tag{4.22}$$

  Choice of α depends on the smoothness requirements. We use $\alpha = {-1}/{2}$, which yields C^1-continuous results. The kernel requires four samples per dimension, and hence is slightly more computationally expensive than the linear kernel.

4.4.4 Pixel Selection

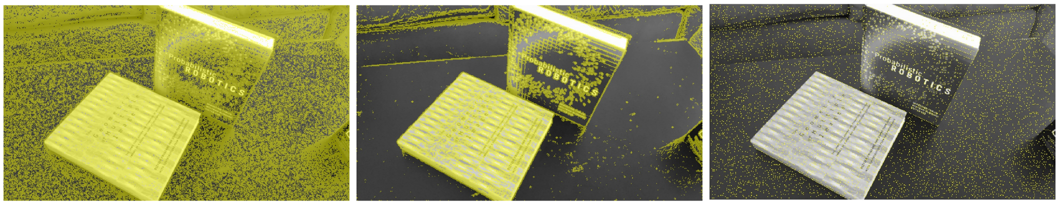
Not all pixels contribute equally to the objective function in Eq. (4.6) as each pixel's contribution is "weighted" by its gradient magnitude. This, in fact, is a good in the sense that no special handling for edges is required [172]. However, vast computational savings could be obtained by reducing the number of pixels. For instance, Dellaert and Collins propose a method that selects the best pixels for image alignment [97]. However, the method requires prior knowledge about the camera motion, which may not be available.

Meilland *et al*. [256] propose a method for pixel selection based on magnitude of the Jacobian per degree-of-freedom. The method can be used to reduce the number of pixels, but it has been found to have little effect on the precision of th estimated parameters [21, 206].

In this work, we experiment with three schemes based on the absolute magnitude of the gradient:

- **None (dense).** Use all pixels with non-zero gradient.
- **Threshold (semi-dense).** Select pixels with an absolute gradient magnitude value greater than a fixed threshold. The threshold is selected so that 30% of the image data are used.
- **Non-maxima suppression (semi-sparse).** Pixels are selected such that their absolute gradient magnitude is a local maxima in a 3×3 neighborhood. The advantage of this strategy is that it ensures an even distribution of pixels across the field-of-view, and does not require hand-tuned thresholds.

For coarser levels of the pyramid, we found no benefit from performing pixel selection. In fact, it is more efficient to process all pixels and avoid the branching logic required for



(a) *None, dense.* *(123800 pixels, 78%)*

(b) *Threshold, semi-dense.* *(44640 pixels, 28%)*

(c) *Non-max suppression.* *(7650 pixels, 4.9%)*

Figure 4.1: *Illustration of the different pixel selection methods. Highlighted areas indicate selected pixels, whose number is shown in parentheses and as percentage of the total number of pixels.*

pixel selection as coarser levels of the pyramid are small enough to fit in higher cache levels. Hence, pixel selection is only applied when the image size is equal to or greater than 320×240. An example of the selected pixels from each the pixel selection method is visualized in Fig. 4.1.

4.4.5 Robust Estimation

To make the objective robust, we rely on Iteratively Re-weighted Least-Squares (IRLS) optimization using an M-Estimator framework [404]. The IRLS procedure has two steps. First, given a vector of residuals $\mathbf{r} \in \mathbb{R}^{n \times 1}$, we determine an estimate of its scale using a robust estimate of the standard deviation given by

$$\hat{\sigma} = 1.4826 \left(1 + \frac{5}{n-6}\right) median_i \left|\mathbf{r}_i\right| . \tag{4.23}$$

The constant 1.4826 is selected to achieve the same efficiency of least-squares under Gaussian noise [404]. The correction factor $5/(n-6)$ is used to account for the effect of small data. In direct VO, the number of residuals n is usually large, on the order of tens of thousands, hence, the term vanishes. It is also possible to use other estimators of scale, such as the Median Absolute Deviation (MAD) [152]. A variety of different robust scale estimates can be found in books by Huber [177] and Maronna et al. [244] and the work by Rousseeuw and Croux [312].

Given the estimate of scale, we now work with normalized residuals given by $x_i = {}^{\mathbf{r}_i}/_{\hat{\sigma}}$. The second step is determining the weights given a choice of the M-Estimator influence function as summarized next.

Given a residual value x, and an influence function $\rho : \mathbb{R} \to \mathbb{R}^+$, the weighting function takes the form

$$w(x) = \frac{\psi(x)}{x}, \text{ with } \quad \psi(x) = \frac{\partial \rho}{\partial x}. \tag{4.24}$$

The two most commonly used influence functions are Huber and Tukey. Huber weights take the form

$$w_{\text{huber}}(x; k) = \begin{cases} \frac{k}{|x|} & \text{if } |x| \geq k; \\ 1 & \text{otherwise,} \end{cases} \tag{4.25}$$

where k is a tuning constant. Tukey's bi-weight function takes the form

$$w_{\text{tukey}}(x; \tau) = \begin{cases} \left(1 - \left(\frac{x}{\tau}\right)^2\right)^2 & \text{if } |x| \leq \tau; \\ 0 & \text{otherwise,} \end{cases} \tag{4.26}$$

where τ is a tuning constant. Tukey's bi-weight treats outliers more aggressively than Huber's and it has been shown to work well in VO [206]. However, it is less smooth and requires more iterations to converge. The two functions are visualized in Fig. 2.6 on Page 34.

In this work, we follow Huber's recommendation [177] and use Eq. (4.25) for weighting across all iteration. Upon convergence at the finest pyramid octave, we repeat the optimization, but this time using Tukey's robust function Eq. (4.26). We found this combination to work well as the optimization takes fewer iterations to converge, and outliers can be rejected entirely. Typically, the optimization will converge after 2–3 additional iterations.

The tuning constants for Huber's and Tukey's functions are respectively $k = 1.345$, and $\tau = 4.6851$ and selected to attain a 95% asymptotic efficiency of the Normal distribution. We refer the reader to the excellent treatment by Zhang [404] for additional details. Finally, the minimizattion of the weighted IC objective is carried out with Gauss-Newton.

4.4.6 Summary

In summary, given a template/reference image $\mathbf{I}$, and an input/moving image $\mathbf{I}'$, the direct VO pipeline consists of the following steps:

- Construct scale-space pyramids from the template and input images. The number of levels is determined such that the minimum image dimension at the coarsest level is at least 40 pixels.

- Pre-compute the Jacobian of the objective (Eq. (4.11)). This may be performed on a subset of pixels given a choice of the pixel selection scheme from Section 4.4.4. Pixel selection is performed when the image size is greater than or equal to 320×240.
- Warp the input image given the current estimate of the parameters $\boldsymbol{\theta}$ using an interpolation scheme from Section 4.4.3 to compute the residuals $\mathbf{I}'(\mathbf{w}(\mathbf{p}; \boldsymbol{\theta})) - \mathbf{I}(\mathbf{p})$.
- Compute the IRLS weights according to Eqs. (4.25) and (4.26) and solve the weighted system to obtain a small update $\Delta\boldsymbol{\theta}$. The update is inverse composed with the current estimate according to Eq. (4.7).
- Repeat the last two steps until convergence. Convergence is determined if the number of iterations exceeds 100, norm of the estimated parameters, or the relative change in the estimates falls below 10^{-6}, the change in the objective function falls below 10^{-6}, or the L^{∞}-norm of the gradient falls below 10^{-8}.

4.5 Experiments

We conduct our experiments on the synthetically generated New Tsukuba dataset [299] as well as a subset of the KITTI benchmark [138]. The synthetic data allows us to isolate the effect of depth estimation errors and focus on the implementation details. Hence, we use the ground-truth disparity maps to initialize the framework.

For the KITTI benchmark, we provide detailed evaluations for the used datasets instead of only reporting average statistics. In the context of this evaluation, detailed statistics in terms of the Absolute Trajectory Error (ATE) and Relative Trajectory Error (RPE) are more informative. Depth estimates are obtained using block matching stereo as implemented in the OpenCV library [53]. We use a window size of 9×9 and a disparity range 0–128.

The framework is implemented in Matlab with some parts accelerated with C++. We use double floating-point precision for all operations, which reduces the effect of roundoff errors. However, as the implementation is in Matlab, an absolute runtime comparison would be biased. Hence, runtime/efficiency is reported in terms of the number of iterations required for convergence. We note that all operations could be optimized to run in real-time by making use of data and instruction level (SIMD) parallelism. In fact, our Matlab implementation runs in excess of 10Hz.

Most VO systems, direct or otherwise, employ a keyframing strategy for computational efficiency and drift reduction. In this work, we use all frames without sub-sampling as we aim to scrutinize the implementation details and observe their effect. Equipping the system with an appropriate keyframing strategy would increase the precision of estimates from each of the design choices.

The direct VO implementation details we evaluate are detailed in Section 4.4. In summary, they are:

- Pixel selection: (1) None, (2) Threshold (**Th**), (3) Non-maxima Suppression (**NMS**).
- Image gradient estimation: (1) Central-differences (**CD**), (2) Central-differences on smoothed images (**CD-s**), (3) Sobel (**SB**), (4) Scharr (**SC**), (5) 5-point stencil (**S5**), (6) Farid and Simoncelli 5-point kernel (**FS-5**), and finally (7) Farid and Simoncelli 7-point kernel [117] (**FS-7**).
- Interpolation: (1) Linear (**Lin**), and (2) Cubic (**Cu**).

In total, there are 42 different permutations of the design choices evaluated in this work. First, we evaluate the accuracy of each of the gradient estimation algorithms in combination with the different interpolation kernels. This is performed with sub-pixel pixel shifts in isolation of the warp parameters.

4.5.1 Accuracy of Linearization

Fundamental to direct VO, and image-based optimization, is the process of linearizing the image in a small neighborhood. Here, we study the accuracy of the different gradient estimation schemes in combination with the different interpolation methods. The 1^{st}-order expansion of an image about small pixel shift $\delta\mathbf{p}$ is given by

$$\mathbf{I}(\mathbf{p}+\delta\mathbf{p}) \approx \mathbf{I}(\mathbf{p}) + \frac{\partial \mathbf{I}}{\partial \mathbf{p}}\delta\mathbf{p}. \tag{4.27}$$

When the magnitude of the sub-pixel shift $\delta\mathbf{p}$ is known, the linearization error can be defined as

$$\mathcal{E}(\delta\mathbf{p}) = \left\|\mathbf{I}(\mathbf{p}+\delta\mathbf{p}) - \mathbf{I}(\mathbf{p}) - \nabla\mathbf{I}\delta\mathbf{p}\right\|^2, \tag{4.28}$$

where $\nabla\mathbf{I} := {}^{\partial\mathbf{I}}/_{\partial\mathbf{p}}$. The accuracy of this approximation depends on two aspects: (1) the interpolation quality, which is required to computed $\mathbf{I}(\mathbf{p}+\delta\mathbf{p})$, and (2) the accuracy of the gradient estimation numerical scheme used to compute $\nabla \mathbf{I}$. Results from the different interpolation and gradient estimation schemes are shown in Fig. 4.2.

As expected, the linearization accuracy is inversely proportional to the magnitude of displacement. Interestingly, for small pixel displacements (≤ 0.5 px), cubic interpolation performs better than linear. However, as the displacement increases, the additional regularity of the cubic interpolation kernel reduces the approximation accuracy. Thus, it may be beneficial for a direct VO system to use linear interpolation at the start of the optimization (and over coarse levels of the pyramid), then apply cubic interpolation for the last few iterations.

The accuracy of the gradient estimation scheme also depends on the magnitude of displacement. For small sub-pixel displacements, central-differences (CD) performs the best. As the displacement increases we see a benefit from smoothing the image prior to differentiation with CD. The benefit of FS-5 and FS-7 starts to appear at larger displacements, but it performs poorly for displacements less than $^1/_2$ a pixel.

Sobel (SB) and Scharr (SC) kernels appear to not have any advantage over central-differences (CD). The 5-point stencil for finite differences (S5) performs better than SB and SC, but not as good as CD.

The evaluation presented here uses translational pixel shifts, which are known to cause systematic biases [330]. In direct VO, pixel displacements are also due to the rotational part of the estimate camera pose, and in fact, the conclusions differ slightly as we show next.

4.5.2 Synthetic Data with Known Depth

To isolate the effect of depth estimation errors and focus on the implementation details, we use the synthetic New Tsukuba dataset, where we initialize the direct VO framework using the provided ground truth disparity maps. We also include baseline results from the excellent feature-based method FOVIS [176], which we also initialize with the ground truth disparity map.

We evaluate performance using the following criteria:

- **Trajectory Root Mean Squared Error (RMS)**, defined as

$$\mathbf{c}_{\text{RMSE}} = \sqrt{\frac{1}{n}\left\|\bar{\mathbf{c}}_i - \hat{\mathbf{c}}_i^2\right\|}, \tag{4.29}$$

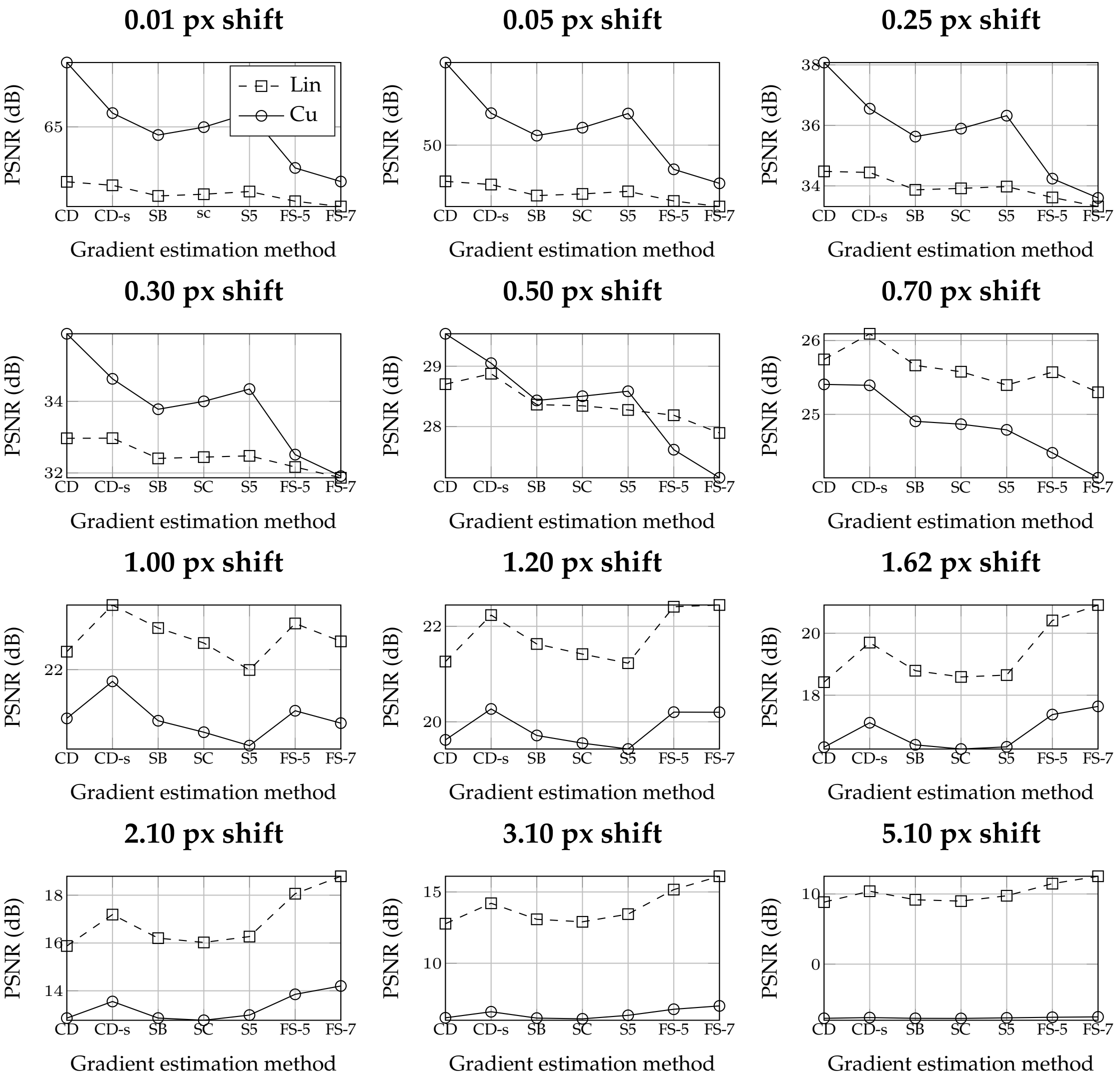


Figure 4.2: Effect of interpolation quality and gradient estimation on the accuracy of linearization. See text for details. Results averaged on many natural images [293].

where $\bar{\mathbf{c}}_i$ is the ground truth cumulative position of the camera, and $\hat{\mathbf{c}}_i$ is the estimated camera position from direct VO. The camera position in the world depends on both the rotation and the translation and is given by

$$\hat{\mathbf{c}}_i = -\hat{\mathbf{R}}_i^\top \hat{\mathbf{t}}_i, \tag{4.30}$$

where $\hat{\mathbf{R}}_i$ and $\hat{\mathbf{t}}_i$ are respectively the rotation and translation estimates at the i^{th} frame.

- **Number of iterations required for convergence** computed at the finest pyramid level. We report the mean and the standard deviation in parentheses.
- **Photometric error** at convergence. Since the synthetic dataset does not contain appearance variations, the photometric error is a direct measure of the quality of the optimization. We also show the mean and standard deviation in parentheses. Note, the photometric is not displayed when evaluating on real data as it is not necessarily a good indication of the optimization performance. This is because of the appearance variations and more challenging lighting conditions when working with real datasets.

Referring to Table 4.1, we observe a consistently higher estimation accuracy in the *dense* setting, where all pixels with non-vanishing gradients are used. We also observe consistently better accuracy with cubic interpolation. The most accurate results are also obtained using the 5-point stencil. The smallest photometric error is achieved using CD. Surprisingly, however, while the photometric error is at its minima, this does not correspond to enhanced estimation accuracy of the camera path. The differentiation filters by Farid and Simoncelli (FS-5, and FS-7) appear to improve the accuracy in the semi-sparse setting. Nonetheless, they lag in performance in comparison to the 5-point stencil. Finally, similar to conclusions in Section 4.5.1, we observe no advantage from using Sobel, or Scharr filters. A graphical depiction of the different methods using cubic interpolation is also shown in Fig. 4.3.

4.5.3 High Frame-Rate Data

We use the high frame-rate dataset by Handa *et al*. [154] to evaluate the different details in the presence of realistic image noise and as a function of inter-frame displacement.

Translational errors are shown in Figs. 4.4 and 4.7 for the noise-free and realistic noise scenarios respectively. Rotational errors are shown in Figs. 4.5 and 4.6 Conclusions are

Table 4.1: *Evaluation of direct VO details on synthetic data. See Section 4.5.2 for details.*

No pixel selection (dense)						
	RMSE		Iterations		Error	
	Lin	Cu	Lin	Cu	Lin	Cu
CD	8.12	7.39	5.94 ± 0.91	4.99 ± 0.77	47.71 ± 21.83	**47.42 ± 22.11**
CD-S	8.17	7.46	5.55 ± 0.83	4.62 ± 0.71	48.68 ± 22.28	48.40 ± 22.56
SB	8.21	7.51	5.70 ± 0.80	4.75 ± 0.70	48.44 ± 22.16	48.15 ± 22.44
SC	8.17	7.46	5.74 ± 0.83	4.80 ± 0.69	48.48 ± 22.18	48.20 ± 22.46
S5	7.98	**7.25**	6.80 ± 1.10	5.72 ± 0.91	48.57 ± 22.22	48.27 ± 22.53
FS-5	8.41	7.74	5.18 ± 0.80	**4.29 ± 0.66**	48.68 ± 22.28	48.38 ± 22.60
FS-7	8.62	7.96	5.14 ± 0.89	4.29 ± 0.79	48.72 ± 22.31	48.41 ± 22.62
Pixel selection with Threshold (semi-dense)						
CD	9.58	8.42	5.22 ± 0.86	4.34 ± 0.62	31.82 ± 13.78	**31.42 ± 14.11**
CD-S	9.63	8.54	4.80 ± 0.92	4.06 ± 0.70	32.17 ± 13.81	31.79 ± 14.15
SB	9.75	8.62	5.01 ± 0.89	4.20 ± 0.65	31.82 ± 13.73	31.43 ± 14.07
SC	9.62	8.51	5.04 ± 0.88	4.21 ± 0.62	32.01 ± 13.81	31.62 ± 14.16
S5	9.33	**8.20**	6.02 ± 0.93	5.06 ± 0.65	32.02 ± 13.95	31.60 ± 14.29
FS-5	10.06	9.00	4.50 ± 1.06	**4.00 ± 1.01**	31.87 ± 13.54	31.51 ± 13.92
FS-7	10.40	9.35	4.58 ± 1.66	4.15 ± 1.75	32.17 ± 13.60	31.83 ± 13.98
Pixel selection with Non-maxima Suppression (semi-sparse)						
CD	11.40	10.88	6.06 ± 2.08	5.16 ± 1.95	10.42 ± 4.78	10.51 ± 4.80
CD-S	9.61	9.22	5.02 ± 1.21	4.36 ± 1.14	11.64 ± 5.51	11.72 ± 5.52
SB	10.98	10.49	5.38 ± 1.53	**4.51 ± 1.43**	10.67 ± 5.01	10.75 ± 5.01
SC	10.08	9.67	5.45 ± 1.34	4.59 ± 1.23	11.65 ± 5.46	11.74 ± 5.47
S5	9.31	**8.96**	7.18 ± 3.66	6.16 ± 1.52	12.64 ± 6.01	12.74 ± 6.02
FS-5	10.23	9.73	4.33 ± 1.87	5.24 ± 3.73	10.68 ± 5.11	10.75 ± 5.12
FS-7	9.72	9.00	6.10 ± 8.15	8.83 ± 13.50	**9.52 ± 4.53**	9.69 ± 4.73

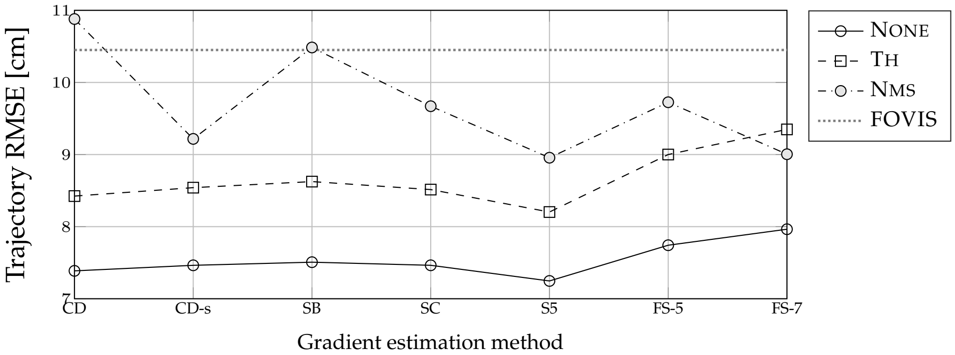


Figure 4.3: *Trajectory RMSE on synthetic data shown as a function of Pixel Selection and Gradient Estimation method. Cubic interpolation for image warping. See Table 4.1 for detailed evaluation.*

similar in both cases, where we the best accuracy is attained using the dense setting combined with CD, CD-s, or S5 for gradient estimation. All plots share the same color scaling for easy comparison.

Number of iterations required for convergence with statistics computed over all pixel selection methods per dataset FPS. For the noise-free case results are shown in Figs. 4.8 and 4.9. Statistics for the realistic noise datasets are shown in Figs. 4.10 and 4.11

4.5.4 Real Data

Evaluation on a subset of the KITTI benchmark is shown in Tables 4.2 and 4.3. We show the translation RPE (t-RPE) reported in cm and the rotation RPE (r-RPE), which we compute at the geodesic distance [179]. The units are degrees, multiplied by 10 for additional details.

Interestingly, results on real data differ from the synthetic data. In general, using all pixels appears to improve the estimates of translation, while selecting a subset of pixels using NMS provides better rotation estimates. Additionally, gradient estimation using FS-5 and FS-7 appear to be better than the alternatives in terms of accuracy. However, FS-7 consistently requires additional iterations for convergence.

The SB and SC methods for gradient estimation appear to not have any advantage over CD. However, due to noise in real data, smoothing the image prior to estimating gradients using CD provides good results. Contrary to the noise-free experiments with synthetic data, S5 is consistently worse.

The effect of interpolation on accuracy agree with results form the synthetic experiments. Due to the large baseline in the KITTI dataset, linear interpolation appears to provide better

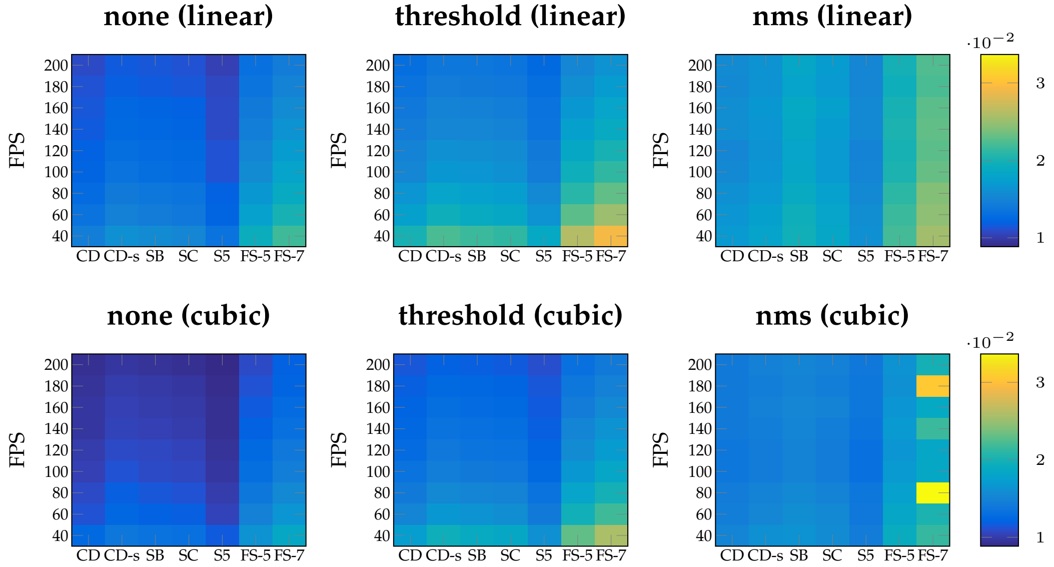


Figure 4.4: Evaluation of translational error on high frame-rate data without image noise.

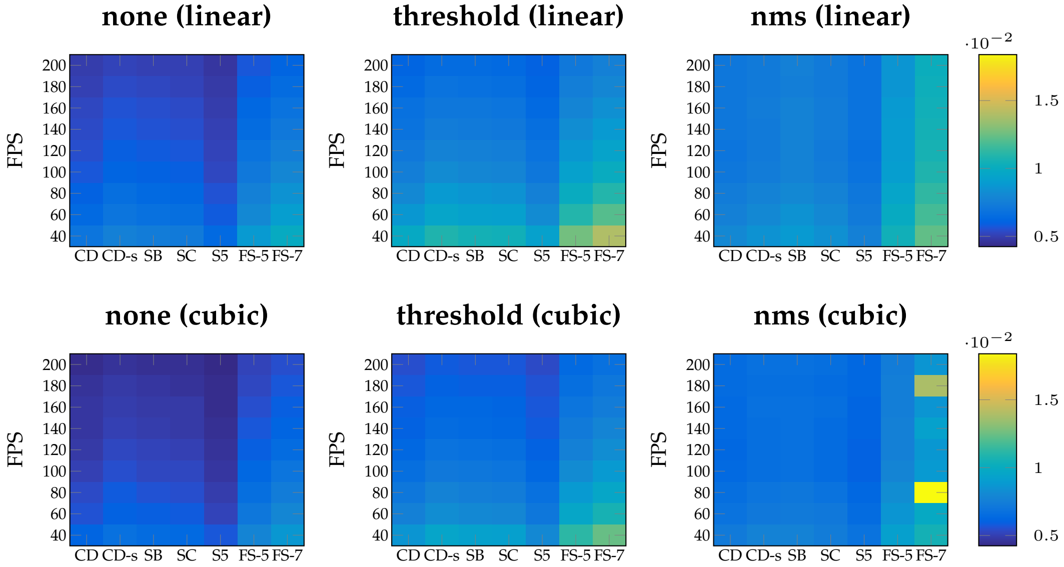


Figure 4.5: Evaluation of rotational error on high frame-rate data without image noise.

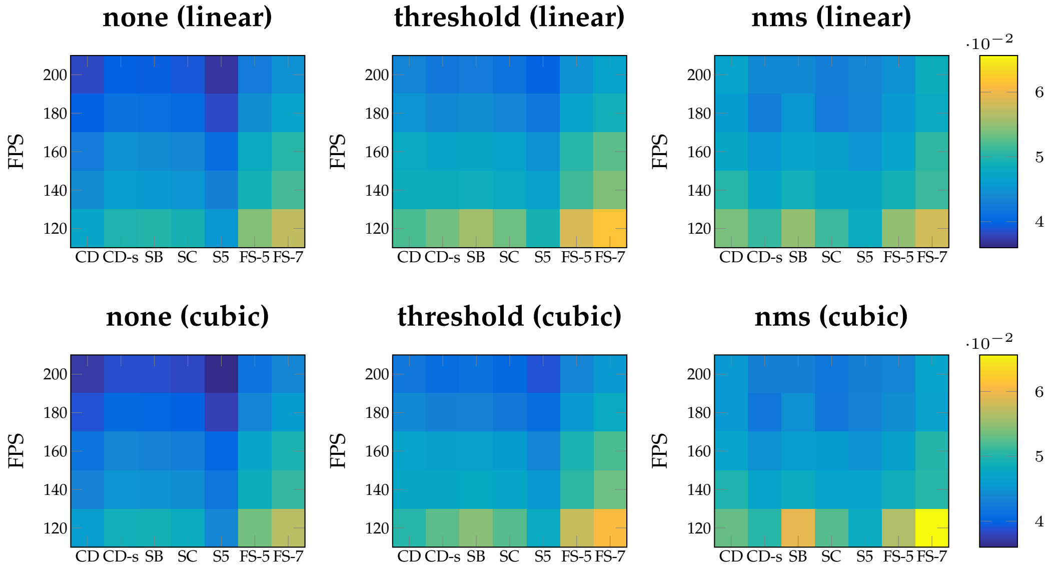


Figure 4.6: *Evaluation of translational error on high frame-rate data with realistic image noise.*

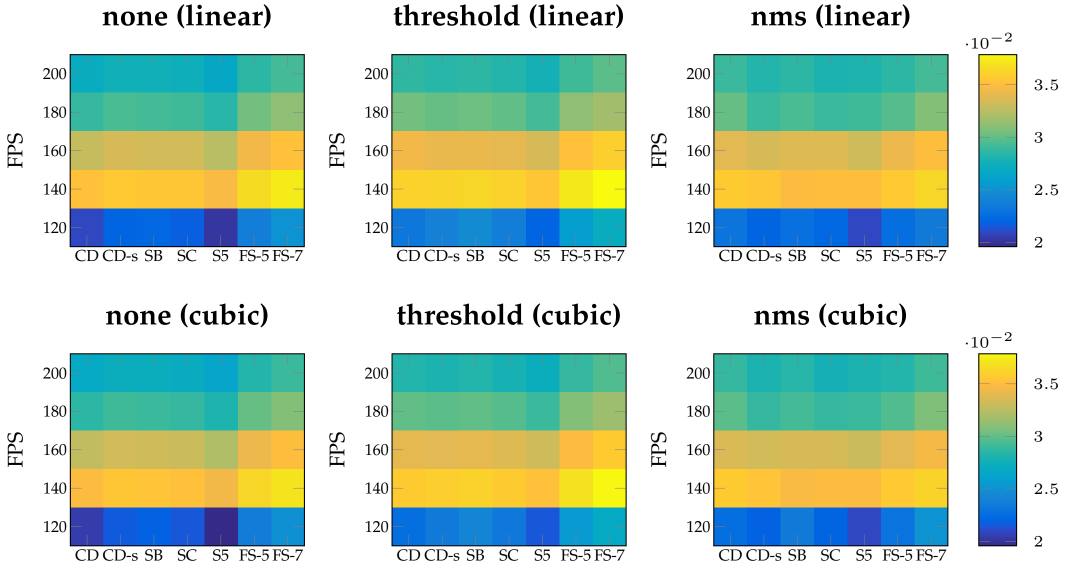


Figure 4.7: *Evaluation of rotational error on high frame-rate data with realistic image noise.*

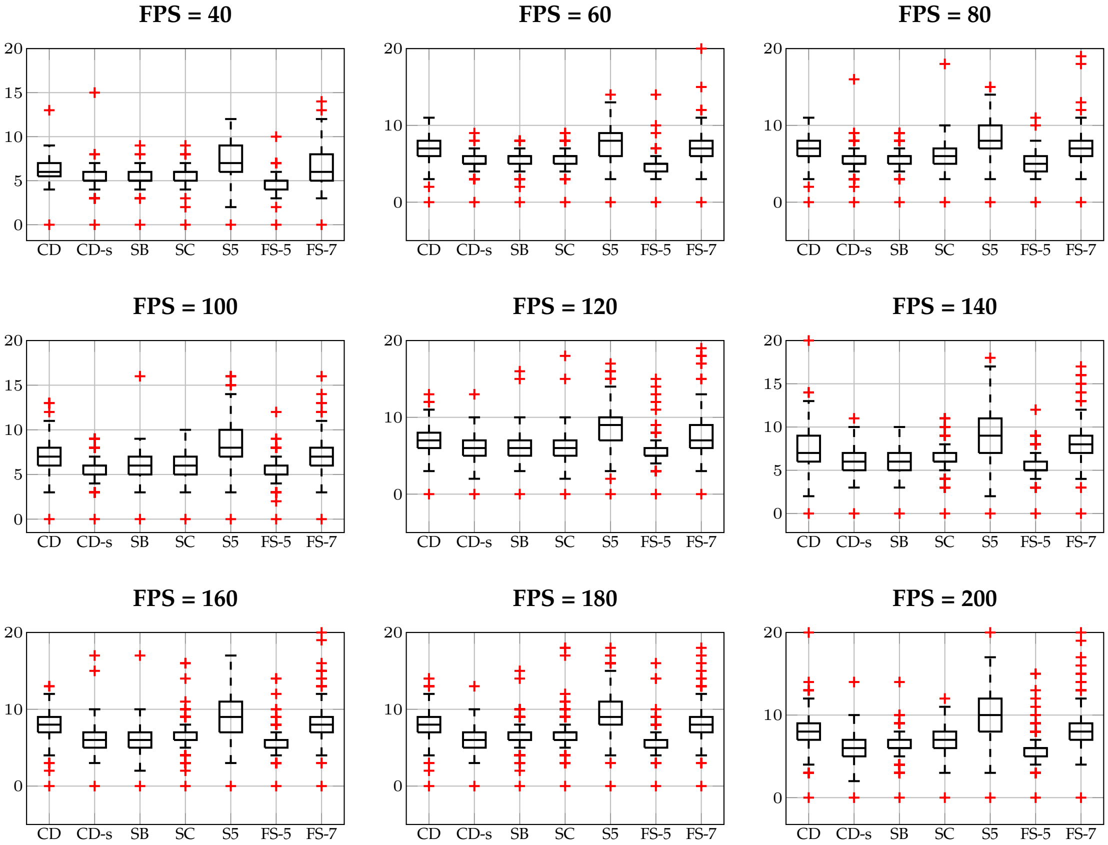
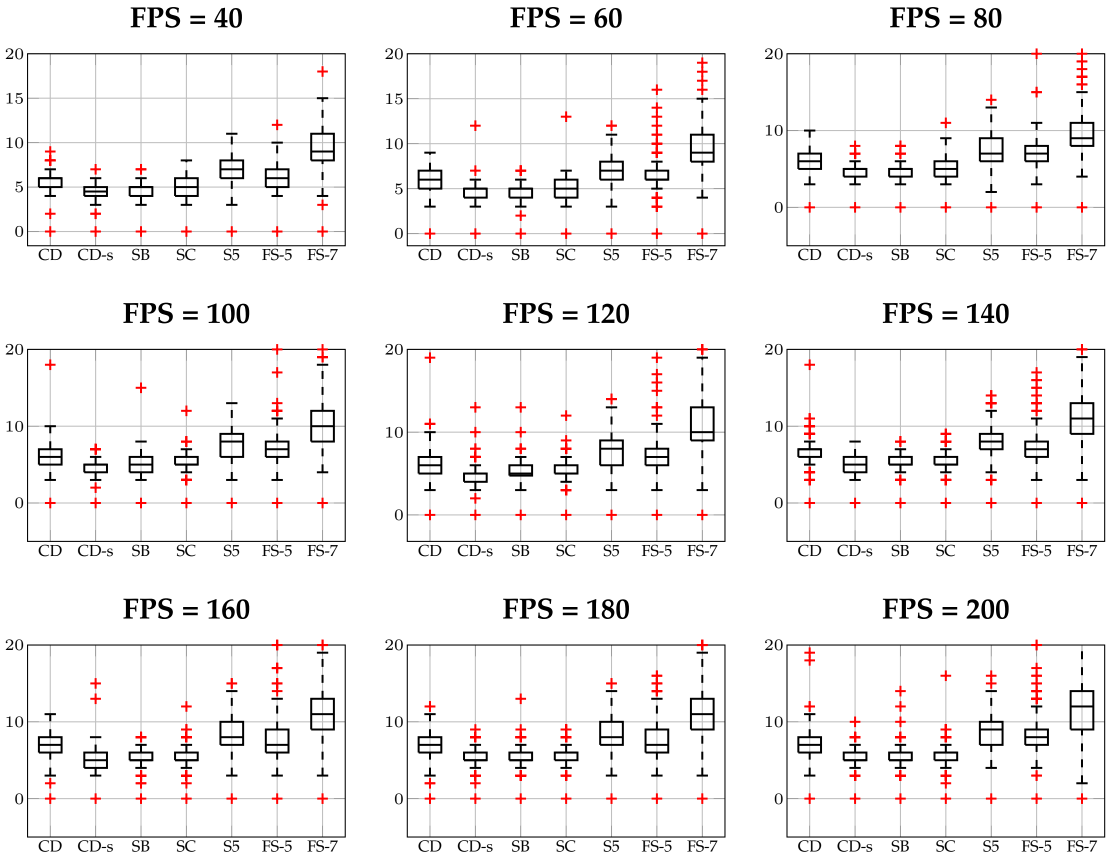


Figure 4.8: *Linear interpolation no noise*


FPS = 40

FPS = 60

FPS = 80

FPS = 100

FPS = 120

FPS = 140

FPS = 160

FPS = 180

FPS = 200

Figure 4.9: *Cubic interpolation no noise*

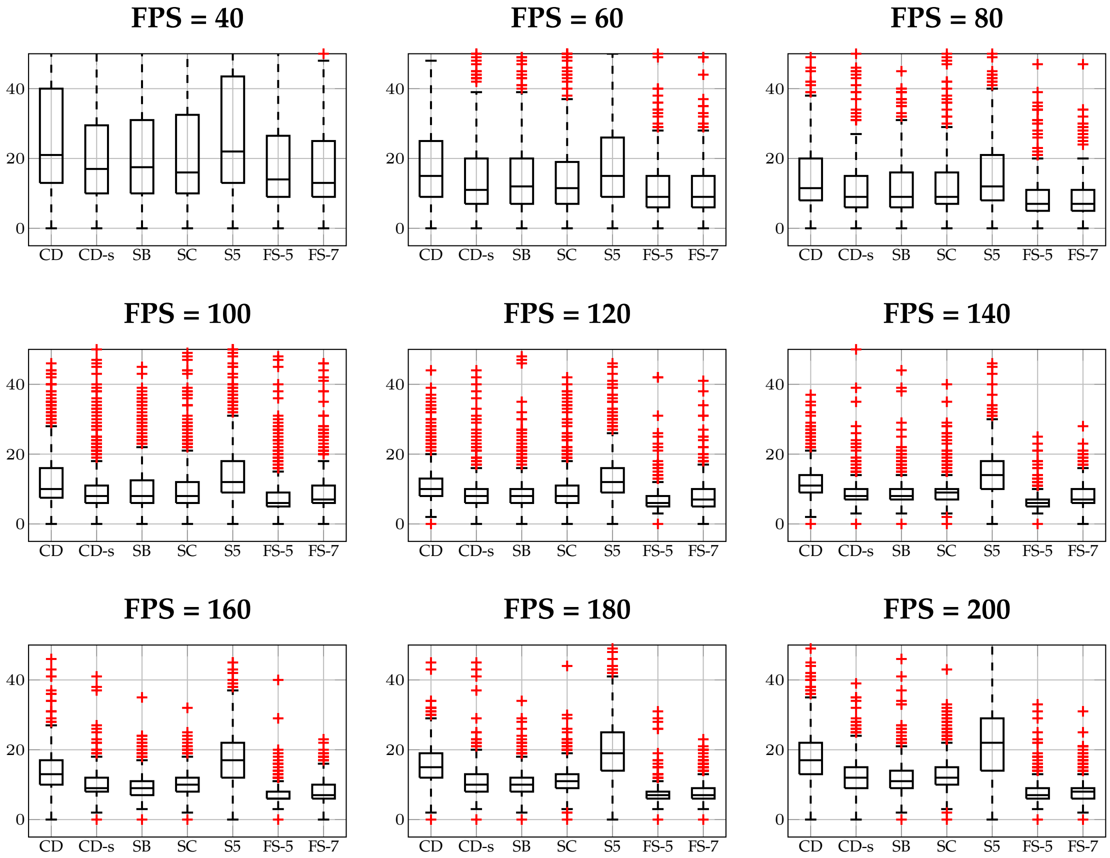


Figure 4.10: *Linear interpolation with noise*

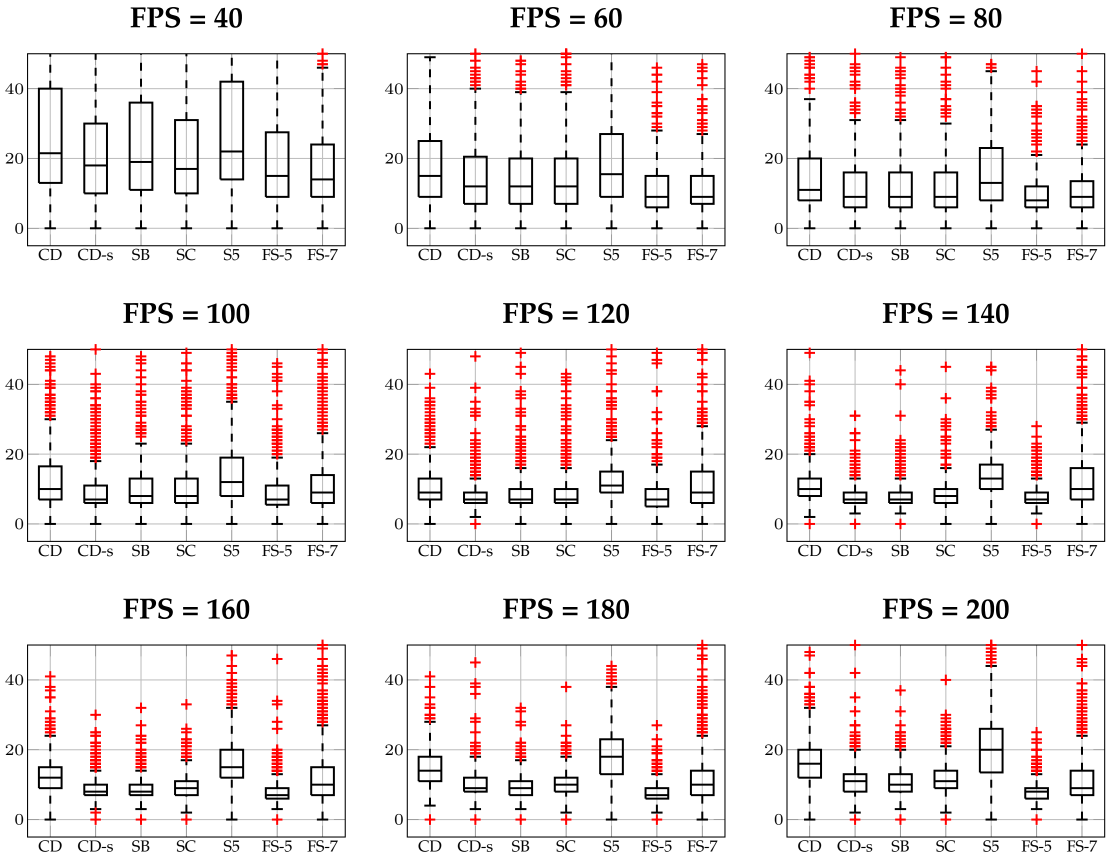


__Figure 4.11:__ Cubic interpolation with noise

results most of the time.

4.6 Summary & Discussion

In this chapter we presented an evaluated of several implementation details at the core of direct VO algorithms. We evaluated the effect of gradient estimation, interpolation and pixel selection. When working with noise-free data with relatively high framerate, the best combination of parameters is: gradient estimation method with the 5-point stencil, using all pixels densly, and cubic interpolation for warping. In addition, when using IRLS, the photometric error in itself is an insufficient metric for estimation quality.

However, when working with noisy real data (with relatively large inter-frame displacement) there is no single best combination. Nonetheless, the optimal filters proposed by Farid & Simoncelli [117] provide the most accurate results. Interestingly, there is no clear benefit from using cubic interpolation to outweigh its computational cost.

Direct VO can produce accurate rotation estimates, but this accuracy is related to the number and distribution of pixels used. A possible cause for this effect is bias towards translation as most of the usable pixels are close to the camera. One could use of this fact to rely on direct VO for rotation estimation and relegating translation estimates to feature-based methods. Or, by separating rotation from translation estimates, possibly based on the estimated depth.

A limitation of direct methods is the reliance on the brightness constancy assumption. The next section presents our work on using feature descriptors to address appearance variations in a direct image alignment framework.

Table 4.2: *Results on KITTI sequence 0*

	t-RMSE		r-RMSE		Iter	
	Lin	Cu	Lin	Cu	Lin	Cu
none						
CD	3.76	3.71	2.37	2.64	8.6 ± 5.1	7.8 ± 4.2
CD-s	3.65	3.44	2.12	1.62	7.8 ± 4.1	7.2 ± 4.1
SB	3.70	3.22	2.12	1.87	8.0 ± 4.3	7.4 ± 4.8
SC	4.16	3.85	2.25	2.58	7.9 ± 3.9	7.3 ± 3.7
S5	5.13	3.88	1.85	2.55	9.3 ± 4.6	8.5 ± 4.3
FS-5	**1.80**	3.22	**1.19**	1.52	7.5 ± 4.7	7.0 ± 4.1
FS-7	1.99	3.49	1.21	1.37	7.5 ± 4.4	7.1 ± 4.2
threshold						
CD	3.75	2.39	2.15	1.90	8.1 ± 4.7	7.5 ± 4.4
CD-s	4.01	3.81	2.50	2.57	7.5 ± 5.1	6.9 ± 4.4
SB	4.02	2.94	2.64	2.64	7.7 ± 4.9	7.0 ± 4.7
SC	3.95	4.43	2.23	2.38	7.8 ± 5.5	7.0 ± 4.5
S5	7.71	4.25	1.99	2.41	9.0 ± 5.4	8.2 ± 4.6
FS-5	4.44	3.90	1.71	2.43	7.1 ± 4.7	6.8 ± 5.3
FS-7	4.08	**1.85**	1.73	**1.18**	7.2 ± 5.2	6.8 ± 3.7
nms						
CD	3.69	3.26	2.26	1.70	8.7 ± 5.8	7.7 ± 5.4
CD-s	3.58	2.88	2.25	1.98	7.2 ± 5.1	6.5 ± 5.3
SB	3.05	3.77	1.63	3.01	7.4 ± 5.6	6.6 ± 5.6
SC	3.78	3.36	1.86	1.98	7.7 ± 6.4	6.7 ± 5.6
S5	4.43	4.15	2.50	2.17	10.1 ± 7.2	8.9 ± 5.3
FS-5	3.93	3.54	1.86	1.84	6.4 ± 5.8	7.3 ± 6.2
FS-7	4.06	3.68	**1.23**	1.24	8.0 ± 7.3	10.8 ± 9.9

Table 4.3: *Results on KITTI sequence 5*

	t-RMSE		r-RMSE		Iter	
	Lin	Cu	Lin	Cu	Lin	Cu
			none			
CD	4.82	5.19	4.47	4.40	8.5 ± 5.7	7.8 ± 5.5
CD-S	3.22	4.48	3.13	3.21	7.8 ± 5.5	7.2 ± 5.2
SB	5.23	4.73	3.19	4.33	7.9 ± 5.5	7.3 ± 5.3
SC	5.36	5.33	3.23	3.22	7.9 ± 5.6	7.2 ± 4.3
S5	5.53	5.53	4.29	4.43	9.6 ± 6.4	8.5 ± 4.9
FS-5	**2.76**	2.82	2.36	3.87	7.3 ± 4.7	6.7 ± 4.5
FS-7	2.82	**2.76**	2.36	2.47	7.3 ± 5.0	6.8 ± 4.4
			threshold			
CD	5.61	5.47	3.34	3.34	8.4 ± 6.4	7.6 ± 5.3
CD-S	3.51	3.22	3.38	3.34	7.7 ± 6.4	7.0 ± 6.1
SB	5.82	5.07	4.57	3.32	8.0 ± 7.1	7.2 ± 5.9
SC	5.15	5.68	3.45	4.52	7.9 ± 6.2	7.2 ± 6.3
S5	6.22	5.76	4.61	4.43	9.4 ± 7.0	8.5 ± 6.3
FS-5	2.94	**2.86**	2.42	2.42	7.2 ± 6.1	6.7 ± 5.6
FS-7	2.95	2.90	2.42	2.43	7.2 ± 6.0	6.8 ± 5.9
			nms			
CD	4.52	5.36	3.28	3.38	8.8 ± 6.9	7.8 ± 6.8
CD-S	3.42	4.94	2.14	3.37	7.3 ± 6.9	6.5 ± 6.5
SB	5.39	5.04	2.34	3.62	7.3 ± 6.2	6.7 ± 7.2
SC	5.16	5.63	2.46	2.39	7.6 ± 7.1	6.8 ± 7.3
S5	6.02	5.78	3.62	3.41	10.1 ± 8.1	8.9 ± 6.9
FS-5	2.89	**2.77**	0.27	2.51	6.3 ± 4.3	7.6 ± 6.4
FS-7	2.84	2.80	**0.25**	**0.26**	8.0 ± 7.4	11.2 ± 9.6

Chapter 5

Bit-Planes: Binary Descriptor for Robust Dense Image Alignment

Contents

5.1 Summary of Contributions

- We study the problem of camera tracking (pose estimation) under sudden and drastic illumination changes.
- We propose an illumination invariant feature descriptor suitable for dense image alignment.
- The descriptor is experimentally shown to be suitable for gradient estimation required by direct methods.
- The descriptor is an adaption of the binary LBP/Census transform to work with the multi-channel Lucas & Kanade algorithm

- The adaption allows minimizing an equivalent form of the Hamming distance in a standard least-squares optimization framework without the need for approximations.

- Due to the compactness of the descriptor, we demonstrate tracking results faster than real-time on mobile devices.

- Code and challenging template tracking datasets are release in open source and can be found at `https://www.cs.cmu.edu/~halismai/bitplanes`.

- Video demonstrations of the approach can be found at `https://goo.gl/fjDzWZ` and `https://goo.gl/nriV2b`.

5.2 Introduction

Binary descriptors are powerful tools for solving *sparse* image alignment problems due to their discriminative power, robustness to illumination change, and low complexity [63, 121, 149, 167, 221, 388]. Matching binary descriptors is typically performed by exhaustive search [51, 196] using binary norms, such as the Hamming distance. Exhaustive search, however, is inefficient when dense correspondences are required in real-time [143, 275] and its accuracy is limited to pixel resolution.

A classical way of speeding up the task of image alignment is to linearize pixel intensities of with respect to geometric displacement. The most notable example of this strategy can be found in the seminal work of Lucas & Kanade (LK) [235]. The LK algorithm aims to establish an approximate linear relationship between appearance and geometric displacements. Efficient linear solvers can then be employed for finding the best alignment parameters with respect to a known template. The relationship between appearance and geometric displacement is seldom linear, so the linearization process is typically repeated until convergence.

At the heart of LK is the notion that an approximate linear relationship between pixel appearance and geometric displacement can be established reliably. Pixel intensities are not deterministically differentiable with respect to geometric displacements. Instead, the linear relationship is established stochastically through spatial finite differences whose outputs we refer to as image gradients. Estimating stochastic gradients on image intensities has a long and rich history dating back to seminal works of computer vision [246]. Furthermore, it has been well documented that pixel intensities of natural images are strongly correlated over small spatial areas further validating the assumed approximate linear relationship between pixel intensities and geometric displacements [337].

Another important requirement in the original LK algorithm is a satisfied brightness constancy assumption (BCA). The BCA states that a pixel's appearance must remain constant after motion. Relying on raw pixel intensities, however, does not always preserve the BCA in most image alignment tasks. This is either due to non-Lambertian scenes, or sudden changes in illumination.

In this work we explore the validity of a *descriptor constancy* assumption using photometrically invariant descriptors in lieu of the commonly employed BCA. In particular, we explore the effectiveness of one of the simplest and most efficient binary descriptors: the original form of Local Binary Patterns (LBP) [290], also known as the Census Transform [400], for robust and efficient dense correspondence estimation problems. The concept of linearizing feature descriptors with respect to geometric displacement within the LK framework is an emerging topic [18, 57, 326]. To date, the descriptors employed in LK have a considerable computational footprint such as HOG [89] and dense SIFT [50, 225] making them unsuitable for in vision applications requiring dense correspondences in real-time under stringent computational requirements.

An important contribution we make in this chapter is to explain, and characterize theoretically, why a naive implementation of the LBP/Census achieves poor performance within the LK algorithm (Section 5.6.2). We argue that a new descriptor must be devised, taking into account the theoretical requirements of the LK algorithm while maintaining the computational advantages of the LBP/Census descriptor. We refer to this LK inspired descriptor herein as bit-planes.

In addition, in order to maintain the invariance properties of binary descriptors, they must be matched under binary norms (*e.g.* the Hamming distance), which are usually neither convex nor continuous. Common strategies to addressing this challenge is to either approximate the binary descriptor with a continuous function [149], or to approximate the binary distance with a smooth form [388], neither of which fully maintain the invariance properties of the descriptor [99]. In contrast, the squared distance between the proposed representation is equivalent to the Hamming distance. Hence, the proposed descriptor retains the illumination invariance properties of whilst using standard least-squares optimization. In the following sections we will:

- Introduce bit-planes, a binary descriptor that can be seemingly used within the LK image alignment framework. Our formulation of the descriptor allows us to minimize the Hamming distance using standard least-squares minimization without resorting to any approximations. Thereby maintaining the illumination invariance of the descriptor.

- Explore the suitability of our descriptor for linearization as a function of geometric displacement. We demonstrate that even though the dense bit-planes descriptor is inherently discontinuous it shares the same critical properties enjoyed by pixel intensities, which make them suitable for efficient gradient-based optimization.

- Discuss the issue of whether pre-computing a binary descriptor in order improve efficiency is viable.

- Evaluate the performance of bit-planes on synthetic image alignment problems to answer fundamental questions about the Descriptor's behavior, such as: suitability of linearization and converges basin. Additional evaluation on real-data and extension to nonlinear warps (visual odometry) is presented in Chapter 6.

We start with a brief summary of the LK algorithm. A more detailed exposition can be found in Section 2.5 of this document and in the excellent series by Baker and Matthews [25], Baker et al. [26, 27, 28, 29].

5.3 The Lucas & Kanade Algorithm

Let $\mathbf{I}_0 : \mathbb{R}^2 \to \mathbb{R}$ be the template/reference image. After camera motion with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$, we obtain an input/moving image $\mathbf{I}_1$. We desire to estimate the parameters of motion such that the following objective is minimized

$$\mathcal{E}(\mathbf{x}; \boldsymbol{\theta}) = \sum_{\mathbf{x} \in \Omega_0} \| \mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}'(\boldsymbol{\theta})) \|_2^2, \tag{5.1}$$

where Ω_0 is a subset of pixels in the template, $\boldsymbol{\theta}$ is an initial estimate of the motion parameters and $\mathbf{x}'(\boldsymbol{\theta})$ describes the transformed pixel coordinates given the motion parameters, commonly known as the *warping* function. By performing a 1^{st} Taylor expansion of Eq. (5.1) in the vicinity of $\boldsymbol{\theta}$, taking the partial derivatives with respect to the parameters, and equating it to zero, we arrive at the *normal equations* given by

$$\mathbf{J}(\mathbf{x}; \boldsymbol{\theta})^\top \mathbf{J}(\mathbf{x}; \boldsymbol{\theta}) \Delta\boldsymbol{\theta} = \mathbf{J}(\mathbf{x}; \boldsymbol{\theta})^\top \mathbf{e}(\mathbf{x}; \boldsymbol{\theta}), \tag{5.2}$$

where $\mathbf{J}(\mathbf{x};\boldsymbol{\theta})$ is the matrix of partial derivatives of the warped image intensities with respect to the motion parameters evaluated at the current estimate of parameters $\boldsymbol{\theta}$, and

$$\mathbf{e}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}'(\boldsymbol{\theta})). \tag{5.3}$$

Using the chain rule, the Jacobian takes the form

$$\mathbf{J}(\mathbf{x};\boldsymbol{\theta}) = \frac{\partial \mathbf{I}_1(\mathbf{x})}{\partial \boldsymbol{\theta}} = \frac{\partial \mathbf{I}}{\partial \mathbf{x}'}\frac{\partial \mathbf{x}'}{\partial \boldsymbol{\theta}}, \tag{5.4}$$

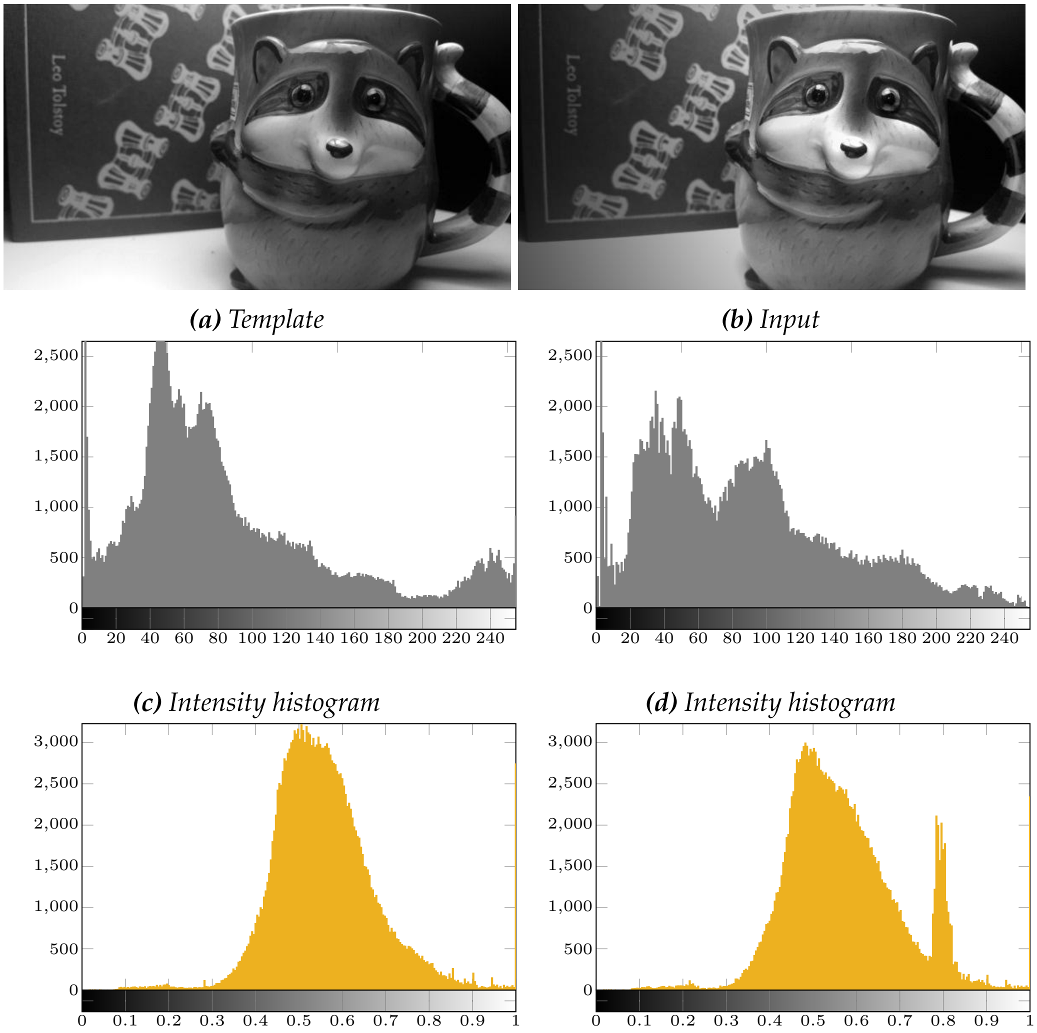
where $\partial \mathbf{I}_1/\partial \mathbf{x}'$ is estimated stochastically through x- and y- finite differences, while $\partial \mathbf{x}'/\partial \boldsymbol{\theta}$ is usually obtained deterministically using the closed-form of the warping function (*cf*. Section 2.2.3). The original formulation of LK is applicable to a variety of problems. For special warps that satisfy a group requirement, however, a more efficient variant is Baker and Matthews' Inverse Compositional algorithm (IC) [25] which we will use in the experimental portion of this paper. A detailed review of the IC algorithm can be found in Section 2.5.2 on Page 38 of this document.

5.3.1 Brightness constancy and photometric variations

The classical formulation of LK relies on the brightness constancy assumption [235], which is seldom satisfied in real life applications. The effect of violations of the brightness constancy are demonstrated in Fig. 5.1, where a nonlinear form of intensity variation is applied to the template image. Although it would seem that the variations are minimal, inspecting the intensity distribution of each of the images indicate otherwise. In contrast, the distribution of our proposed descriptor remain close to its original shape prior to the application of nonlinear intensity deformations.

Techniques to address illumination change are discussed Section 2.7.5 and include: (i) estimating illumination parameters alongside the motion parameters (either jointly [36] or in an alternating fashion [376]), (ii) using intrinsically robust similarity measures, such as Mutual Information [90, 103], or the normalized correlation [112, 180], and (iii) preprocessing the images to obtain an illumination robust representation [18, 236, 390].

On the one hand, estimating illumination is sensitive to the modeling assumptions and is difficult to craft correctly. On the other hand, optimizing robust metrics is sensitive to the initial conditions and requires general purpose optimizers that cannot exploit the special structure of least-squares problems [289] (see Section 2.4 for an explanation of the special structure of least-squares problems, and [289, ch. 10]).



***(a)** Template*

***(b)** Input*

***(c)** Intensity histogram*

***(d)** Intensity histogram*

***(e)** Proposed descriptor histogram*

***(f)** Proposed descriptor histogram*

***Figure 5.1:** Although nonlinear variations in appearance may appear subtle, a closer lock at the distribution of intensities indicate otherwise. The shape of the distribution of the proposed descriptor remain close to its distribution prior to intensity deformation.*

Preprocessing the image does not typically require restrictive assumptions. Traditionally, preprocessing an image is done by convolving with filters, or other simple operations such as whitening the signal [124, 352]. Densely sampled feature descriptors are another form of preprocessing, which we adopt in this work. In particular, we propose the use of the bit-planes descriptor. During evaluation, we show that our approach exceeds the robustness of algorithms that explicitly model illumination as well as methods that rely on robust cost metrics. Furthermore, our method is more efficient, and simpler to implement. Central to our work is the extension of LK to multi-channel images, which we review next.

5.3.2 Multi-channel LK

Extending LK to multi-channel images is straightforward, it is presented here to introduce notation. Let $\boldsymbol{\phi}_0 : \mathbb{R}^2 \to \mathbb{R}^d$ be the d-channel representation of the template/reference image. Employing a similar notation to the classical LK algorithm, after camera motion with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$, we obtain an input/moving d-channel representation $\boldsymbol{\phi}_1$. To align descriptors using LK we seek to minimize:

$$\mathcal{E}_{\boldsymbol{\phi}}(\mathbf{x}; \boldsymbol{\theta}) = \sum_{\mathbf{x} \in \Omega_0} \|\boldsymbol{\phi}_0(\mathbf{x}) - \boldsymbol{\phi}_1(\mathbf{x}'(\boldsymbol{\theta}))\|^2. \tag{5.5}$$

To linearize Eq. (5.5) we must obtain an estimate of the Jacobian

$$\mathbf{J}_{\boldsymbol{\phi}}(\mathbf{x}; \boldsymbol{\theta}) = {}^{\partial \boldsymbol{\phi}}\!/\!_{\partial \boldsymbol{\theta}} \in \mathbb{R}^{d \times p}. \tag{5.6}$$

Let the value of the j-th channel, as illustrated in Fig. 5.2, of the multi-channel representation be denoted as $\boldsymbol{\phi}^j(\mathbf{x})$, where

$$\boldsymbol{\phi}(\mathbf{x}) = [\boldsymbol{\phi}^1(\mathbf{x}) \dots \boldsymbol{\phi}^d(\mathbf{x})]^\top. \tag{5.7}$$

The sought Jacobian for each channel in Eq. (5.5) can be obtained using the chain rule

$$\frac{\partial \boldsymbol{\phi}_1^j(\mathbf{x})}{\partial \boldsymbol{\theta}} = \frac{\partial \boldsymbol{\phi}_1^j}{\partial \mathbf{x}'} \frac{\partial \mathbf{x}'}{\partial \boldsymbol{\theta}} \text{ for } j = 1, \dots, d \tag{5.8}$$

where ${}^{\partial \boldsymbol{\phi}_1^j}\!/\!_{\partial \mathbf{x}'}$ is estimated stochastically through x- and y- finite difference filters on $\boldsymbol{\phi}_1^j$, and ${}^{\partial \mathbf{x}'}\!/\!_{\partial \boldsymbol{\theta}}$ is obtained deterministically from the closed-form of warping function. The multi-channel $d \times p$ Jacobian matrix can then be formed as

$$\mathbf{J}_{\boldsymbol{\phi}}(\mathbf{x};\boldsymbol{\theta}) = \frac{\partial \boldsymbol{\phi}_1(\mathbf{x})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \partial \phi_1^1(\mathbf{x})/\partial \boldsymbol{\theta} & \dots & \partial \phi_1^d(\mathbf{x})/\partial \boldsymbol{\theta} \end{bmatrix}^\top \tag{5.9}$$

Using this multi-channel linearization all extensions and variations of the LK algorithm can be used with different multi-channel descriptors. Recent work has demonstrated the utility of multi-channel LK using classical dense descriptors such as dense SIFT, HOG LBP, and variations on LBP [18, 57].

5.4 Dense Binary Descriptors

Local Binary Patterns (LBP) [290] were among the first binary descriptors proposed in vision. An identical representation was independently developed by Zabih & Woodfill under the name: Census Transform (CT) [400], which remains popular in stereo and optical flow [149, 278, 342, 388] due to its efficiency and illumination invariance properties.

LBP is based on the predicate of pixel comparisons in a small neighborhood as illustrated in Fig. 5.3. By definition, the LBP descriptor is invariant to monotonic illumination changes given that it is matched under a binary norm. Recently, binary descriptor research has progressed significantly with the development of several high performance descriptors such as ORB [63] and BRISK [221] among others [32, 63, 223, 356]. All such descriptors may be used as *bit-planes*, but we chose a LBP evaluated in a 3×3 neighborhood for its efficiency and locality.

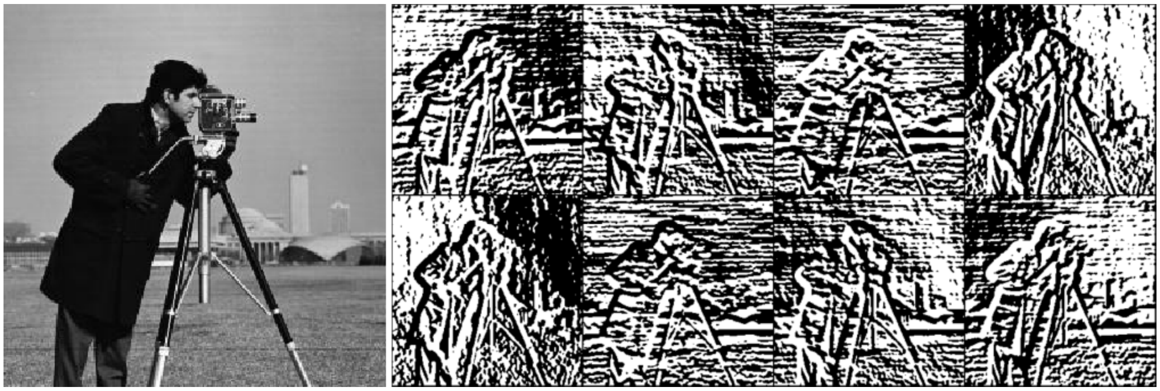
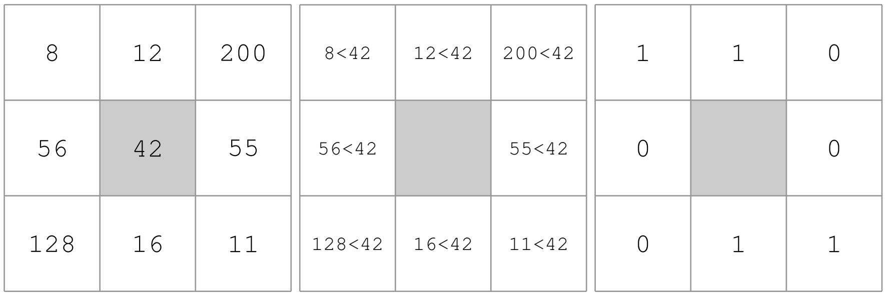


Figure 5.2: *An example of the LBP descriptor evaluated on a 3×3 neighborhood, which results in an 8-channel bit-planes descriptor.*



8	12	200
56	42	55
128	16	11

(a)

8<42	12<42	200<42
56<42		55<42
128<42	16<42	11<42

(b)

1	1	0
0		0
0	1	1

(c)

Figure 5.3: *The canonical LBP descriptor is obtained by performing pixel comparisons in a fixed order and converting the binary string to a decimal value. In Fig. 5.3a the center pixel is compared to its neighbors as shown in Fig. 5.3b. The descriptor is obtained by combining the results of each comparison in Fig. 5.3c into a single scalar signature.*

5.4.1 Single channel LBP descriptor

When extracting LBP about a pixel position $\mathbf{x}$ one obtains,

$$\phi(\mathbf{x}) = \sum_{i=1}^{8} 2^{i-1} \big[\mathbf{I}(\mathbf{x}) \bowtie \mathbf{I}(\mathbf{x} + \Delta\mathbf{x}_i) \big], \tag{5.10}$$

where $\{\Delta\mathbf{x}_i\}_{i=1}^{8}$ is the set of the eight relative coordinate displacements possible within a 3×3 neighborhood around the center pixel location $\mathbf{x}$. Other neighborhood sizes and sampling locations can be used, but we found a 3×3 region to perform best. The operator $\bowtie \in \{>, \geq, <, \leq\}$ is a pixel comparison/binary test, and the bracket denotes the indicator function. We refer to the LBP descriptor described in Eq. (5.10) as single-channel since its output is a scalar at every pixel position $\mathbf{x}$. A visual depiction of the single-channel LBP descriptor estimation process is shown in Fig. 5.3.

5.4.2 Bit-planes descriptor

When matching binary descriptors, such as LBP, it is common practice to employ the Hamming distance. This is important because the Hamming distance is invariant to the ordering of pixel comparisons within the neighborhood used to compute the descriptor. In contrast, the sum or squared distances (SSD) lacks this desirable property and is dependent on the ordering specified by $\{\Delta\mathbf{x}_i\}_{i=1}^{8}$. This becomes problematic when employing dense binary descriptors within the multi-channel LK framework due to its inherent dependence on the SSD. To make dense binary descriptors compatible with LK we propose the *bit-planes* descriptor given by:

$$\phi(\mathbf{x}) = \left[\mathbf{I}(\mathbf{x}) \bowtie \mathbf{I}(\mathbf{x} + \Delta\mathbf{x}_1) \quad \ldots \quad \mathbf{I}(\mathbf{x}) \bowtie \mathbf{I}(\mathbf{x} + \Delta\mathbf{x}_8) \right]^\top \in \mathbb{R}^{8\times 1} \,. \tag{5.11}$$

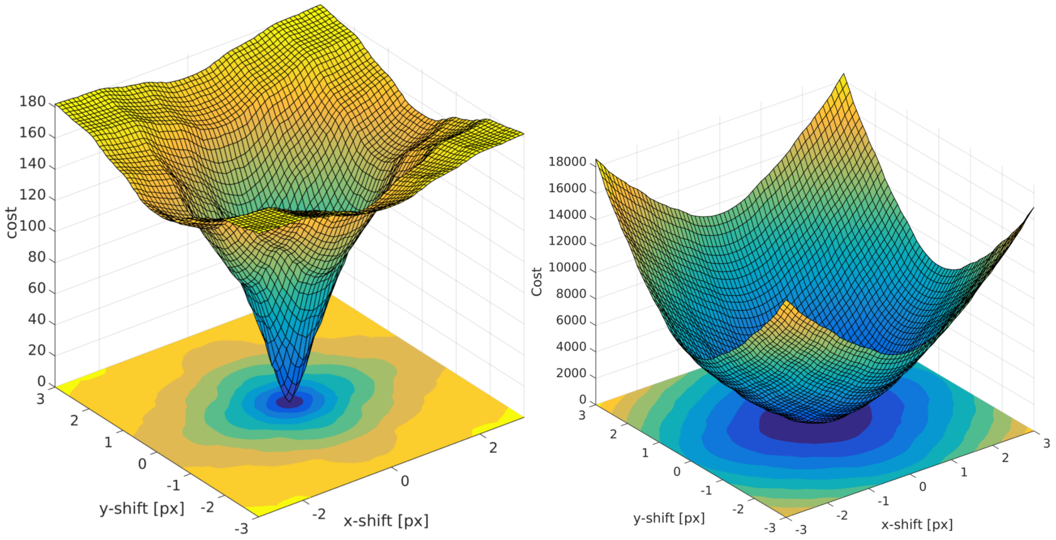
For each pixel coordinate $\mathbf{x}$ in the image, this descriptor produces an 8-channel binary valued vector. Notably, using the SSD with the multi-channel representation in Eq. (5.11) between two bit-planes descriptors is equivalent to the Hamming distance between single-channel LBP descriptors. Specifically, the ordering of the pixel comparisons within the 3×3 neighborhood of the bit-planes descriptor has no effect on the SSD.

The Hamming distance is defined as the sum of mismatched bits between two binary strings [151]. To illustrate the equivalence between the Hamming distance and the sum of squared errors using bit-planes we use an example composed of three bits. Let $\mathbf{a} = \{1, 0, 1\}$, and $\mathbf{b} = \{0, 1, 1\}$. The Hamming distance between $\mathbf{a}$ and $\mathbf{b}$ is 2 as the bit strings differ at two locations. The sum of squared differences between $\mathbf{a}$ and $\mathbf{b}$ is given by $(1-0)^2 + (0-1)^2 + (1-1)^2$, which is the same as the Hamming distance.

5.5 Linearizing Bit-Planes

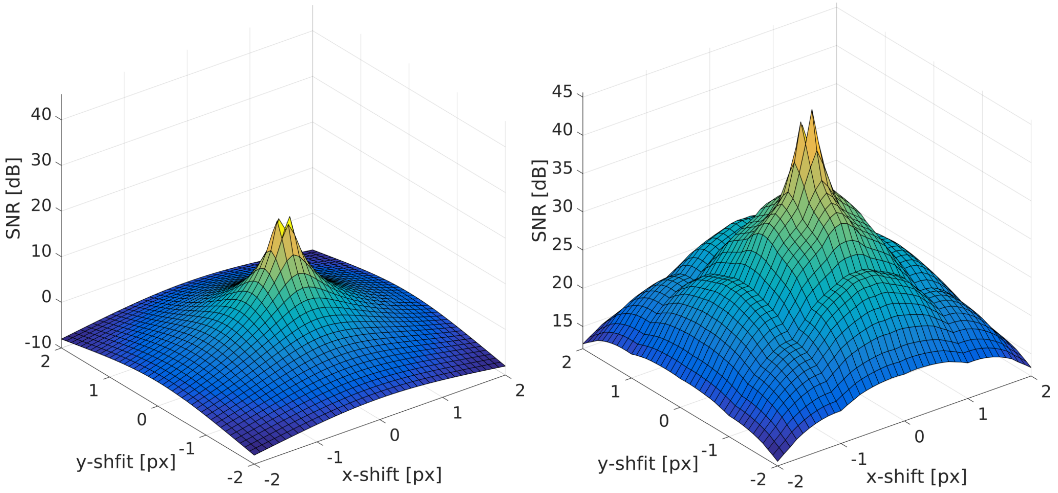
In order for the bit-planes descriptor to be effective within a multi-channel LK framework we first need to ensure existence an approximate linear relationship between the bit-planes and geometric displacements. Inspecting a visualization of the bit-planes descriptor in Fig. 5.2, one could be doubtful about the existence of such relationship as each channel of the descriptor is highly discontinuous. In addition, estimating stochastic gradients per binary channel seems strange as they can take on only a handful of possibilities (*e.g.* using central differences, and excluding the image border, the gradient values can be either 0 or $\pm 1/2$).

The news is not all gloomy. In Fig. 5.4b we see the SSD cost surface between a patch within a natural image and shifted versions of itself in the x- and y- directions averaged over a subset of natural images. As expected, we observe the quasi-convex cost surface for raw pixel intensities. The shape of the cost surface is important to the effectiveness of the LK algorithm — as the LK objective relies on a graceful reduction of the SSD cost as a function of geometric displacement. Interestingly, when inspecting Fig. 5.4a we see a similar quasi-convex cost surface, albeit not as wide, which indicates that bit-planes have similar properties to raw pixel intensities when using the SSD as a measure of dissimilarity.



(a) *Bit-planes.* ***(b)*** *Raw intensity.*

Figure 5.4: *Cost surface of our bit-planes descriptor Fig. 5.4a computed over a subset of natural images [399] in comparison to the SSD over raw intensity Fig. 5.4b. Both cost surfaces are suitable for LK.*



(a) *Bit-planes.* ***(b)*** *Raw intensity.*

Figure 5.5: *Assessment of the linearization properties of the bit-planes descriptor in terms of the signal-to-noise-ratio (SNR) as a function of translational displacement. Even though raw pixels are superior in this context, bit-planes offer a sufficient approximation to be used within a gradient-based optimization framework.*

Consider a translational displacement warp $\Delta\boldsymbol{\theta} \in \mathbb{R}^2$ where we attempt to linearly predict an image representation $\mathbf{R}$ (raw pixels $\mathbf{I}$, or bit-planes $\boldsymbol{\phi}$) in the x- and y- directions,

$$\mathbf{R}(\mathbf{x}(\mathbf{0})) + \frac{\partial \mathbf{R}(\mathbf{0})}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} \approx \mathbf{R}(\mathbf{x}(\Delta\boldsymbol{\theta})). \tag{5.12}$$

The error of this linear approximation is given by

$$\boldsymbol{\epsilon}(\Delta\boldsymbol{\theta}) = \sum_{\mathbf{x}\in\Omega} \|\mathbf{R}(\mathbf{x}(\mathbf{0})) + \frac{\partial \mathbf{R}(\mathbf{0})}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} - \mathbf{R}(\mathbf{x}(\Delta\boldsymbol{\theta}))\|_2^2 , \tag{5.13}$$

and its signal-to-noise-ratio (SNR) can be computed using

$$\text{SNR}(\Delta\boldsymbol{\theta}) = 10 \cdot \left(\log \sum_{\mathbf{x}\in\Omega} \|\mathbf{R}(\mathbf{x}(\mathbf{0}))\|_2^2 - \log \boldsymbol{\epsilon}(\Delta\boldsymbol{\theta}) \right) . \tag{5.14}$$

In Fig. 5.5 we depict the SNR of the linearized objective as a function of increasing translational shifts from the true minima for both raw intensities, and bit-planes. The experiments were carried out similarly through the use of a subset of natural images and aggregated to form the results in Fig. 5.5. As expected, the SNR when using binary features is lower than using raw intensities due to the additional quantization when using binary data. However, it seems that — at least qualitatively — bit-planes gradient estimates provide a good local linear approximation of the objective. Hence, further justifying the use of the bit-planes descriptor within the LK framework.

5.6 Experiments

In this section we answer a number of important questions regarding the validity of the dense bit-planes descriptor for robust and efficient image alignment.

5.6.1 Pre-computing descriptors

An obvious question to ask when considering the application of multi-channel descriptors within the LK framework is: whether we can pre-compute the descriptors before warping? Specifically, due to the iterative nature of the LK algorithm it becomes computationally expensive to re-compute the descriptor after each image warping step. Substantial efficiencies can be integrated into any LK-based image alignment if one can pre-compute the descriptor before warping. This is illustrated in Fig. 5.6.

We answer this question in Fig. 5.8 where we evaluated a number of LK variants [25]: forward additive (FA), forward compositional (FC), and inverse compositional (IC) for the task of image alignment on natural images using random warps and including nonlinear appearance variations of the form

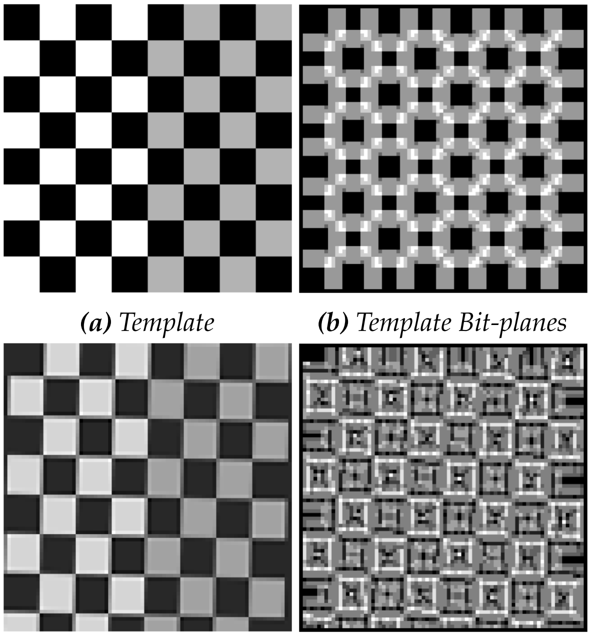
$$\mathbf{I}_1(\mathbf{x}) = 255\left(\frac{\alpha \mathbf{I}_0(\boldsymbol{\theta}_a(\mathbf{x})) + \beta}{255}\right)^{1+\gamma}, \tag{5.15}$$

where $\boldsymbol{\theta}_a\left(\cdot\right)$ are the 6DOF parameters of an affine warp, α and β are multiplicative and additive lighting change terms, and $|\gamma| < 1$ is used for gamma correction. As expected, we observe that warping feature channels is less accurate than re-computing the descriptor on the warped image as shown in Fig. 5.8.

The degree to which warping the feature channels vs. re-computing them affects accuracy depends on the application and the type of warp. For simple warps such as 2D translation, the relationship between intensity deformation as a function of warp parameters is linear. Hence, approximating multi-channel LK by warping the feature channels is equivalent to re-computing the features on warped images as shown in Fig. 5.9. However, for more complicated warps where deformation of image intensities is nonlinearly dependent on the warp parameters we expect a pronounced difference in alignment accuracy. This is because the value of each descriptor channel might significantly differ after a nonlinear warp. Overall, it is possible to approximate the multi-channel objective in Eq. (5.5) with warping feature channels depending on the type and accuracy requirements of the application at hand. In our experiments, we chose to recompute descriptors after every iteration of image warping.

5.6.2 Single Channel LBP within LK

Employing bit-planes requires the alignment of eight separate channels as opposed to a single channel when working with raw intensities. In Section 6.6 we discussed the problems of using a LBP descriptor within the LK framework. In particular, the representation is inherently sensitive to the ordering of pixel comparisons when using a SSD measure of dissimilarity. Using LBP descriptors within a LK framework as been reported to perform well [149, 388] given small displacements. However, under moderate displacements, the use of the LBP descriptor in LK introduces biases due to choices of the binary test and neighborhood ordering. In Fig. 5.11 we show the effect of differing binary comparison



(a) *Template* ***(b)*** *Template Bit-planes*

(c) *Warped template* ***(d)*** *Warped Bit-planes*

Figure 5.6: *A synthetic image to demonstrate the difference between warping the descriptors vs. re-computing it. Simply warping a pre-computed descriptor does not match the expected output. Alignment error for this example is show in Fig. 5.7. Bit-planes is visualized by summing the eight channels.*

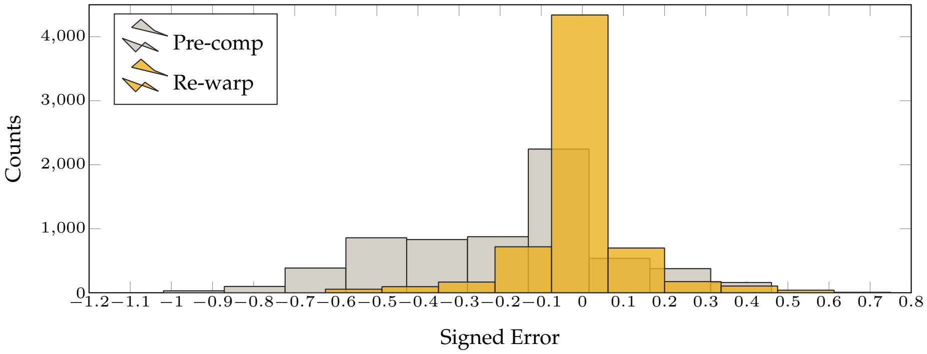


Figure 5.7: *Histogram of error corresponding to warping pre-computed descriptors (absolute mean error* 0.25 ± 0.2*, absolute median error* 0.19*) vs. re-computing the descriptors on warped image (absolute mean* 0.02 ± 0.11 *absolute median* 0*).*

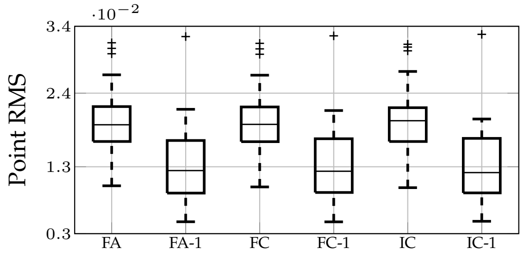


Figure 5.8: *Recomputing descriptors* after *image warping shows (indicated with the suffix '-1') is more accurate than warping the descriptors.*

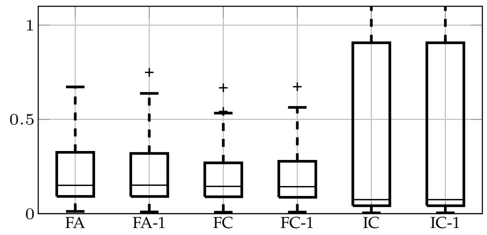


Figure 5.9: *No significant difference between recomputing the descriptors vs. warp the channels when the warp is linear (2d translation).*

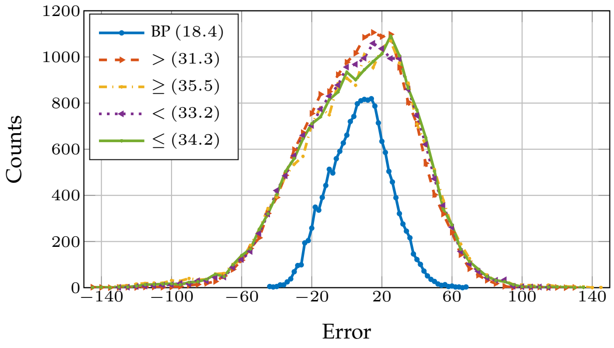


Figure 5.10: *Histogram of intensity errors when using our bit-planes* (BP) *vs. classical single-channel LBP descriptors with different comparison operators. The RMS is shown in parentheses.*

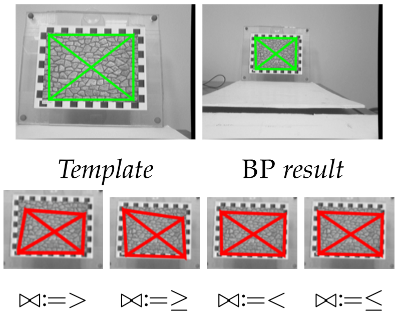


Figure 5.11: *Drift when using LBP vs. bit-planes. The bottom row shows the result of template tracking using LBP. Images are magnified for better visualization (compare with* BP *result). Best viewed in color.*

operators $\bowtie \in \{>, \geq, <, \leq\}$ compared to our proposed bit-planes descriptor. The evaluation is performed on a benchmark dataset by Gauglitz et al. [136], where we see that the bit-planes descriptor is unaffected by the ordering. In our experiments we noticed indistinguishable differences in performance between binary comparison operators when employing the bit-planes descriptor. As a result, we chose to use the $>$ operator for the rest of our experiments.

5.7 Summary

In this chapter, we presented the bit-planes descriptor; an adaption of the LBP/Census transform suitable for nonlinear gradient-based optimization. Not only that bit-planes maintains the invariance to monotonic changes enjoyed by binary descriptors, but also is suitable for dense alignment with subpixel accuracy.

The main limitation of the bit-planes descriptor is the narrow basin of convergence in comparison to using raw pixel intensities. This, however, is expected as there is a natural trade off between robustness and other performance aspects.

In the next chapter, we extend the idea of multi-channel alignment of feature descriptors to different local descriptors. We will also evaluate the performance of the descriptor on challenging pose estimation problems that include: robust and real-time planar template tracking under sudden and drastic illumination change, as well as robust visual odometry with difficult illumination.

Chapter 6

Robust Pose Estimation with Densely Evaluated Descriptors

Contents

6.1 Summary of Contributions

- We evaluate the performance of pose estimation (image alignment) using densely evaluated feature descriptors using our proposed descriptor (bit-planes) as well as a number of other suitable feature descriptors for direct camera tracking.
- Suitability of descriptors for dense alignment is established as a function of the descriptor's spatial locality.
- Performance of image alignment using feature descriptors is evaluated on a two problems: Affine and planar template tracking, as well as visual odometry.

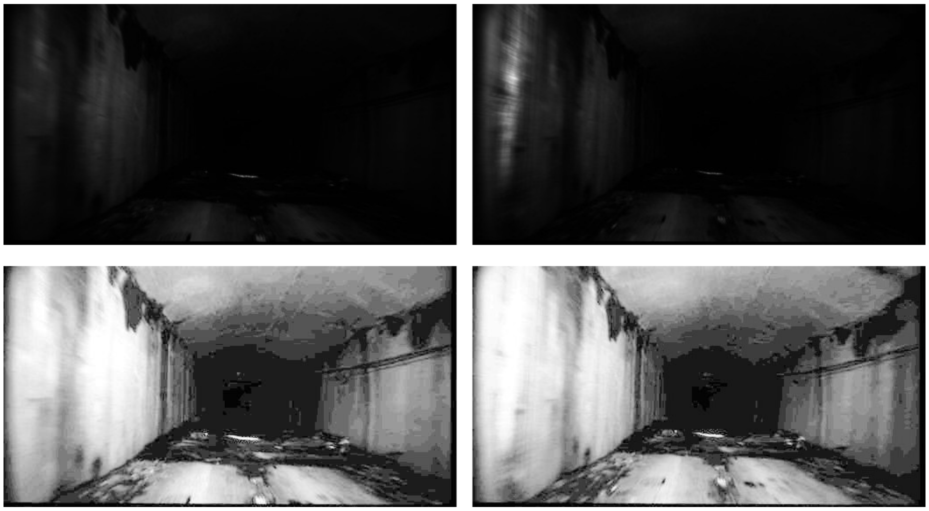


Figure 6.1: Shown at the top row are two consecutive images collected from an underground mine. The bottom row shows a histogram equalization of the images for better visualization. The equalized images may appear noiseless due to its smooth appearance in the document (due to resizing). The data has a low signal-to-noise ratio due to the poor illumination of the scene.

- For template tracking, we demonstrate robust performance in face of sudden and drastic illumination changes, where we outperform the state-of-the-art in terms of robustness and run-time. A video demonstration is available at https://www.youtube.com/watch?v=6d5_IGAoKW0.
- The bit-planes descriptor is shown to run faster than real-time (400+ fps on a laptop, and 100+ fps on mobile devices).
- Open source implementation of the visual odometry system is available in open source at https://github.com/halismai/bpvo, which includes optimized implementations of bit-planes and the rest of the descriptors compared in this chapter.
- A video demonstration of the robust VO algorithm using bit-planes in underground mines is available at https://www.youtube.com/watch?v=fEddznFo3aY.

6.2 Introduction

With the increasing availability of high frame rate cameras, direct tracking is becoming a more popular tool in myriad applications such as visual odometry [75, 199], visual SLAM [110, 154], augmented and virtual reality [316] and dense reconstruction [285]. Advantages of direct tracking include:

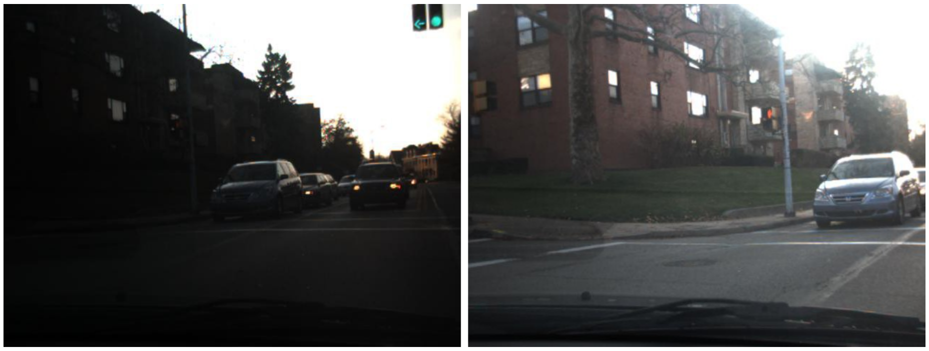


Figure 6.2: *An example of the nonlinear intensity deformation caused by the automatic camera settings. A common problem with outdoor applications of robot vision.*

(i) Increased precision as much of the image could be used to estimate a few degrees of freedom [181].

(ii) Enhanced tracking robustness in feature-poor environments, where high frequency image content (corners and edges) are not readily available.

(iii) Improved ability in handling ambiguously textured scenes [128].

(iv) Improved running time by exploiting the trivially parallel nature of direct tracking [285].

However, as discussed in Chapter 5, the main limitation of direct tracking is the reliance on the *brightness constancy* assumption [173, 235], which is seldom satisfied in the real world. Since the seminal works of Lucas and Kanade [235] and Horn and Schunk [173], researchers have been actively seeking more robust tracking systems [36, 43, 61, 103, 112, 278]. Nevertheless, the majority of research efforts have been focused on two ideas: One, is to rely on intrinsically robust objectives, such as maximizing normalized correlation [180], or the Mutual Information [103], which are inefficient to optimize and more sensitive to the initialization point [289]. The other, is to attempt to model the illumination parameters of the scene as part of the problem formulation [36], which is usually limited by the modeling assumptions.

In this chapter, we study the use of **densely evaluated local feature descriptor as a non-parametric means to achieving illumination invariant dense tracking**. We will show that while feature descriptors are inherently discontinuous, they are suitable for gradient-based optimization when used in a multi-channel framework. We will also show that, depending on the feature descriptor, it is possible to tackle challenging illumination conditions without resorting to illumination modeling assumptions, which are difficult to craft correctly. Finally, we show that the changes required to make use of local feature descriptors in current tracking systems are minimal. Furthermore, the additional computational cost is not a significant barrier.

There exists a multitude of previous work dedicated to evaluating direct tracking. For instance, Baker and Matthews [25] evaluate a range of linearization and optimization strategies along with the effects of parameterization and illumination conditions. Handa *et al*. [154] characterize direct tracking performance in terms of the frame rate of the camera. Klose *et al*. [206] examine the effect of different linearization and optimization strategies on the precision of RGB-D direct mapping. Zeeshan *et al*. [407] explore the parameter space of direct tracking considering power consumption and frame rate on desktop and mobile devices. Sun *et al*. [352] evaluate different algorithms and optimization strategies for optical flow estimation. While Vogel et al. [388] evaluate different data costs for optical flow. Nonetheless, the fundamental question of the quantity being optimized, especially the use feature descriptors in direct tracking, has not yet been fully explored.

Feature descriptors, whether hand crafted [260], or learned [211], have a long and rich history in Computer Vision and have been instrumental to the success of many vision applications such as Structure-from-Motion (SFM) [369], Multi-View Stereo (MVS) [132] and object recognition [338]. Notwithstanding, their use in direct tracking has been limited and is only beginning to be explored [18, 56]. One could argue that this line of investigation has been hampered by the false assumption that feature descriptors, unlike pixel intensities, are non-differentiable due to their discontinuous nature. Hence, the use of feature descriptors in direct tracking has been neglected from the onset.

Among the first application of descriptors in direct tracking is the “distribution fields” work [325, 326], which focused on preserving small image details that are usually lost in coarse octaves of the scale space. Application of classical feature descriptors such as SIFT [234] and HOG [89] to Active Appearance Models have been also explored in the literature demonstrating more robust alignment results [18]. The suitability of discrete feature descriptors for the linearization required by direct tracking has also been investigated in recent work [57], where it was shown that if feature coordinates are independent, then gradient estimation of feature channels can be obtained deterministically using finite difference filters. This is advantageous as gradient-based optimization is more efficient and more precise than discrete optimization [225]. Recent work have applied local descriptors to template tracking [83] in an effort to track non-Lambertian surfaces more robustly.

This chapter serves as a generalization of the bit-planes descriptor presented in Chapter 5, where we experiment with different feature descriptors and evaluate the method on real datasets with ground truth.

6.3 Robust Parametric Image Alignment

In Section 2.5 (Page 35) of this document we provide additional details on image alignment and associated algorithm. In the following, however, we reintroduce the important parts.

Let the intensity of a pixel coordinate $\mathbf{p} = (u,\ v)^\top$ in the *reference* image be given by $\mathbf{I}(\mathbf{p}) \in \mathbb{R}$. After camera motion, a new image is obtained $\mathbf{I}'(\mathbf{p}')$. The goal of direct tracking is to estimate an increment of the camera motion parameters $\Delta\boldsymbol{\theta} \in \mathbb{R}^d$ such that the photometric error is minimized

$$\Delta\boldsymbol{\theta}^* = \underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathbf{p}\in\Omega} \left\| \mathbf{I}'\left(\mathbf{w}(\mathbf{p}; \boldsymbol{\theta} \boxplus \Delta\boldsymbol{\theta})\right) - \mathbf{I}\left(\mathbf{p}\right) \right\|^2, \tag{6.1}$$

where Ω is a subset of pixel coordinates of interest in the reference frame, $\mathbf{w}\left(\cdot\right)$ is a *warping* function that depends on the parameter vector we seek to estimate, and $\boldsymbol{\theta}$ is an initial estimate of the motion parameters. After every iteration, the current estimate of parameters is updated (*i.e.* $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \boxplus \Delta\boldsymbol{\theta}$), where $\boxplus$ generalizes the addition operator over the optimization manifold. The process is repeated until convergence, or some termination criteria have been satisfied [25, 235].

By (conceptually) interchanging the roles of the template and input images, Baker & Matthews devise a more efficient alignment techniques known as the Inverse Compositional (IC) algorithm [25]. Under the IC formulation we seek an update $\Delta\boldsymbol{\theta}$ that satisfies

$$\Delta\boldsymbol{\theta}^* = \underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathbf{p}\in\Omega} \|\mathbf{I}'\left(\mathbf{w}(\mathbf{p}; \boldsymbol{\theta})\right) - \mathbf{I}\left(\mathbf{w}(\mathbf{p}; \Delta\boldsymbol{\theta})\right)\|^2. \tag{6.2}$$

The optimization problem in Eq. (3.4) is nonlinear irrespective of the form of the warping function or the parameters, as — in general — there is no linear relationship between pixel coordinates and their intensities. By equating the partial derivatives of the first-order Taylor expansion of Eq. (3.4) to zero, we reach at solution given by the following closed-form (normal equations)

$$\Delta\boldsymbol{\theta} = \left(\mathbf{J}^\top\mathbf{J}\right)^{-1}\mathbf{J}^\top\mathbf{e}, \tag{6.3}$$

where $\mathbf{J} = \left(\mathbf{g}(\mathbf{p}_1)^\top,\ \ldots,\ \mathbf{g}(\mathbf{p}_m)^\top\right) \in \mathbb{R}^{m\times d}$ is the matrix of first-order partial derivatives of the objective function, m is the number of pixels, and $d = |\boldsymbol{\theta}|$ is the number of parameters.

Each $\mathbf{g}$ is $\in \mathbb{R}^{1\times d}$ and is given by the chain rule as

$$\mathbf{g}(\mathbf{p})^\top = \nabla \mathbf{I}(\mathbf{p}) \frac{\partial \mathbf{w}}{\partial \boldsymbol{\theta}}, \tag{6.4}$$

where $\nabla \mathbf{I} = \left({\partial \mathbf{I}}/{\partial u},\ {\partial \mathbf{I}}/{\partial v} \right) \in \mathbb{R}^{1\times 2}$ is the image gradient along the u- and v- directions respectively. The quantity

$$\mathbf{e}(\mathbf{p}) = \mathbf{I}'(\mathbf{w}(\mathbf{p}; \boldsymbol{\theta})) - \mathbf{I}(\mathbf{p}) \tag{6.5}$$

is the vector of residuals. Finally, the parameters are updated via the IC rule given by

$$\mathbf{w}\left(\mathbf{p}, \boldsymbol{\theta}\right) \leftarrow \mathbf{w}\left(\mathbf{p}, \boldsymbol{\theta}\right) \circ \mathbf{w}\left(\mathbf{p}, \Delta\boldsymbol{\theta}\right)^{-1}. \tag{6.6}$$

6.3.1 Direct Tracking with Feature Descriptors

Direct tracking using image intensities (the brightness constraint in Eq. (7.8)) is known to be sensitive to illumination change. To address this limitation, we propose the use of a *descriptor constancy* assumption. Namely, we seek an update to the parameters such that

$$\Delta\boldsymbol{\theta}^* = \underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \left\| \boldsymbol{\phi}(\mathbf{I}'\left(\mathbf{w}(\mathbf{p}; \boldsymbol{\theta} \boxplus \Delta\boldsymbol{\theta})\right)) - \boldsymbol{\phi}(\mathbf{I}\left(\mathbf{p}\right)) \right\|^2, \tag{6.7}$$

where $\boldsymbol{\phi}(\cdot)$ is a multi-dimensional feature descriptor applied to the reference and the warped input images.

The descriptor constancy objective in Eq. (6.7) is more complicated than its brightness counterpart in Eq. (6.1) as feature descriptors are high dimensional and the suitability of their linearization remains unclear. In the sequel, we will show that various descriptors linearize well and are suitable for direct tracking.

6.3.2 Desiderata

The usual goal of direct tracking is to maximize the precision of the estimated parameters. The linearization required in direct tracking implicitly assumes that we are close enough to the local minima. This fact is usually expressed by assuming small displacements between the input images. In order to maximize precision, it is important to balance the complexity of the descriptor as a function of its sampling density. Namely, descriptors with long range spatial connections such as SIFT [234] and HOG [89], while robust to a range of deformations in the image, they contribute little to tracking precision. This is due to the increased dependencies between pixels contributing to the linear system. We will experimentally validate this hypothesis in the experimental section. Hence, good descriptors for illumination invariant tracking must be:

(1) locally limited with respect to their spatial extent, and

(2) efficient to compute, which is desired for practical reasons as the estimation process is iterative.

Both requirements, locality and efficiency, are closely related as most local descriptors are efficient to compute as well.

6.3.3 Pre-computing descriptors for efficiency

Descriptor constancy as stated in Eq. (6.7) requires re-computing the descriptors after every iteration of image warping. In Chapter 5, we established that our binary descriptor (bit-planes) performance is more accurate when the descriptors are re-computed post image warping. When using bit-planes, however, re-computing the descriptor is computationally cheap.

Nonetheless, re-computing descriptors is undesirable in situations where

- The descriptor requires a non-trivial computational time.
- Warping individual pixels is challenging due to their dependence on extraneous, potentially sparse, depth information, *e.g.* vision-based SLAM. The difficulty arises from the lack of a 3D model that could be used to reason about occlusions, and discontinuities in the image.

Due to the computational and technical limitations of re-computing the descriptor, it is desirable to warp the feature-channels instead of re-computing them after every iteration. Hence, an approximation to the descriptor constancy objective in Eq. (6.7) is to pre-compute the descriptors and minimize the following expression instead:

$$\min_{\Delta\boldsymbol{\theta}} \sum_{\mathbf{p}\in\Omega} \sum_{i=1}^{N_c} \|\boldsymbol{\Phi}'_i(\mathbf{w}(\mathbf{p};\boldsymbol{\theta} \boxplus \Delta\boldsymbol{\theta})) - \boldsymbol{\Phi}_i(\mathbf{p})\|^2, \tag{6.8}$$

where $\mathbf{\Phi}_i$ indicates the i-th coordinate of the pre-computed descriptor and N_c is the number of channels. An illustration of this concept is show in Fig. 5.6.

Although this approximation is less accurate, we found that the loss of accuracy induced when using Eq. (6.8) instead of Eq. (6.7) insignificant in comparison to the computational savings and simplicity of implementation.

To this end, we will consider various feature descriptors in the literature that are suitable for high-precision illumination invariant tracking. Dense feature descriptors used in this work are evaluated in the next section.

6.4 Densely Evaluated Descriptors

We evaluate a number of descriptors suitable for dense tracking as summarized in Table 6.1. Below, we include a brief description of each of the descriptors. A visualization of each of the descriptors is shown in Fig. 6.3.

- **Raw intensity**: this is the trivial form of a feature descriptor, which uses the raw image intensities. We work with grayscale images, and hence it is a single channel, $\mathbf{\Phi}(\mathbf{I}) = \{\mathbf{I}\}$.

- **Gradient constancy**: the image gradient measures the rate of change of intensity and hence it is invariant to additive changes [61]. We found that including the raw image intensities in the optimization with the gradient constraint to work better. The descriptor is composed of three channels and is given by: $\boldsymbol{\phi} = \{\mathbf{I},\ \nabla_u \mathbf{I},\ \nabla_v \mathbf{I}\}$

- **Laplacian**: the Laplacian is based on the 2nd order derivatives of the image and, similar to the gradient constraint, it provides invariance to additive change, but using only a single channel. We found that including the raw intensities to improve results. The descriptor is given by: $\mathbf{\Phi} = \left\{\mathbf{I},\ |\nabla^2 \mathbf{I}|\right\}$.

- **Descriptor Fields (DF) [83]** where the idea is to separate the image gradients into different channels based on their sign. After that, a smoothing step is performed. Using first-order image gradients, denoted by DF-1, is composed of four channels and is given by: $\mathbf{\Phi}_{\text{DF-1}}(\mathbf{I}) = \left\{[\nabla_u \mathbf{I}]^+,\ [\nabla_u \mathbf{I}]^-,\ [\nabla_v \mathbf{I}]^+,\ [\nabla_v \mathbf{I}]^-\right\}$. The 2nd order DF, denoted by DF-2, includes 2-nd order gradient information and is composed of 10 channels. Note, the DF approach is not a binary/octal representation.

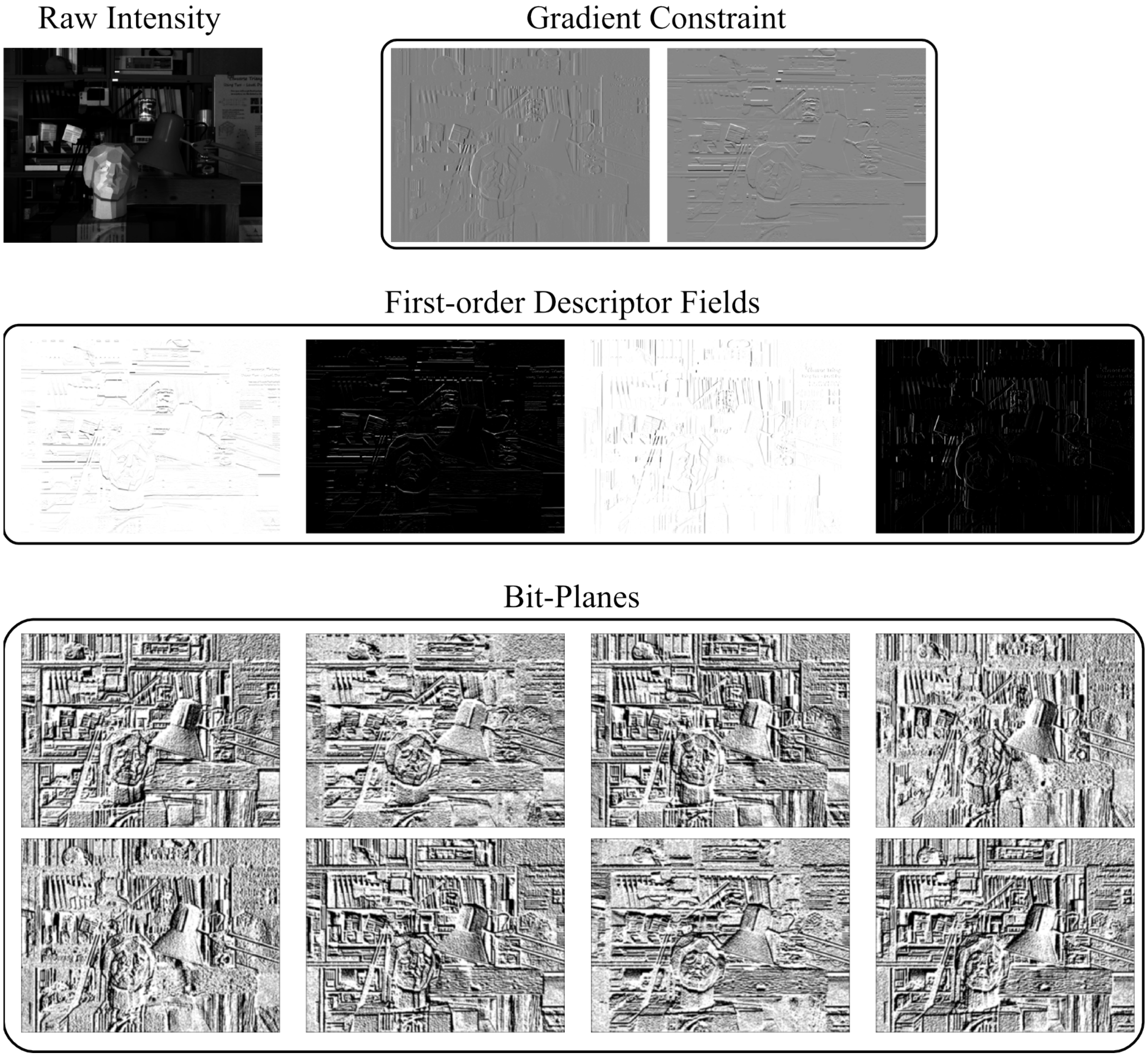


Figure 6.3: *Visualization of the different descriptors. Best viewed on a screen. Note, the Descriptor Fields is not binary.*

Table 6.1: *Descriptors evaluated in this work.*

Name	Acronym	channels
Raw Intensity	RI	1
Gradient Constraint	GC	3
Laplacian	LP	2
1^{st} order DF [83]	DF-1	4
2^{nd} order DF [83]	DF-2	10
Bit-Planes [14]	BP	8

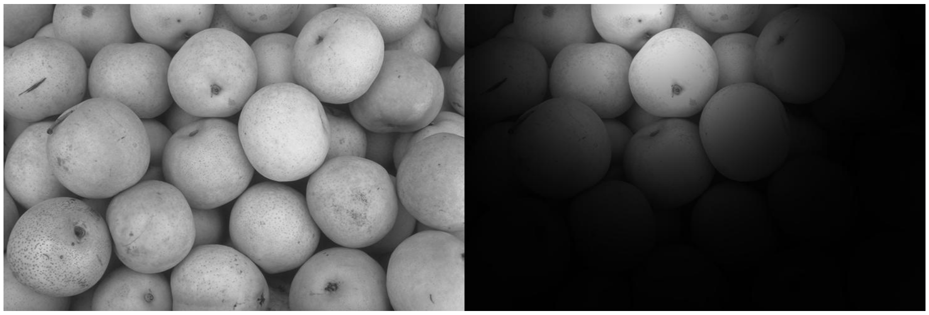


Figure 6.4: *Example illumination change according to Eq.* (6.11).

- **Bit-Planes**: the descriptor we introduced in Chapter 5, where channels are constructed by performing local pixel comparisons. When evaluated in a 3×3 neighborhood, the descriptor results in eight channels given by: $\boldsymbol{\Phi}(\mathbf{I}) = \left\{\mathbf{I}(\mathbf{x}) \geq \mathbf{I}(\mathbf{x} + \Delta\mathbf{x}_j)\right\}_{i=1}^{8}$, where $\mathbf{I}(\mathbf{x} + \Delta\mathbf{x}_j)$ indicates the image sampled at the j-th neighbor location in the neighborhood.

The descriptors we evaluate here are efficient and simple to implement. A MATLAB implementation of each is available in Appendix A. Hence, it is easy to integrate them into existing image alignment systems.

6.5 Evaluation under Controlled Settings

We experiment with the different feature descriptors summarized in Table 6.1 on two direct tracking problems. The first is parametric motion estimation using an Affine motion model, which we use to illustrate performance on synthetically controlled illumination variations. The second is a direct visual odometry approach, which is more challenging as

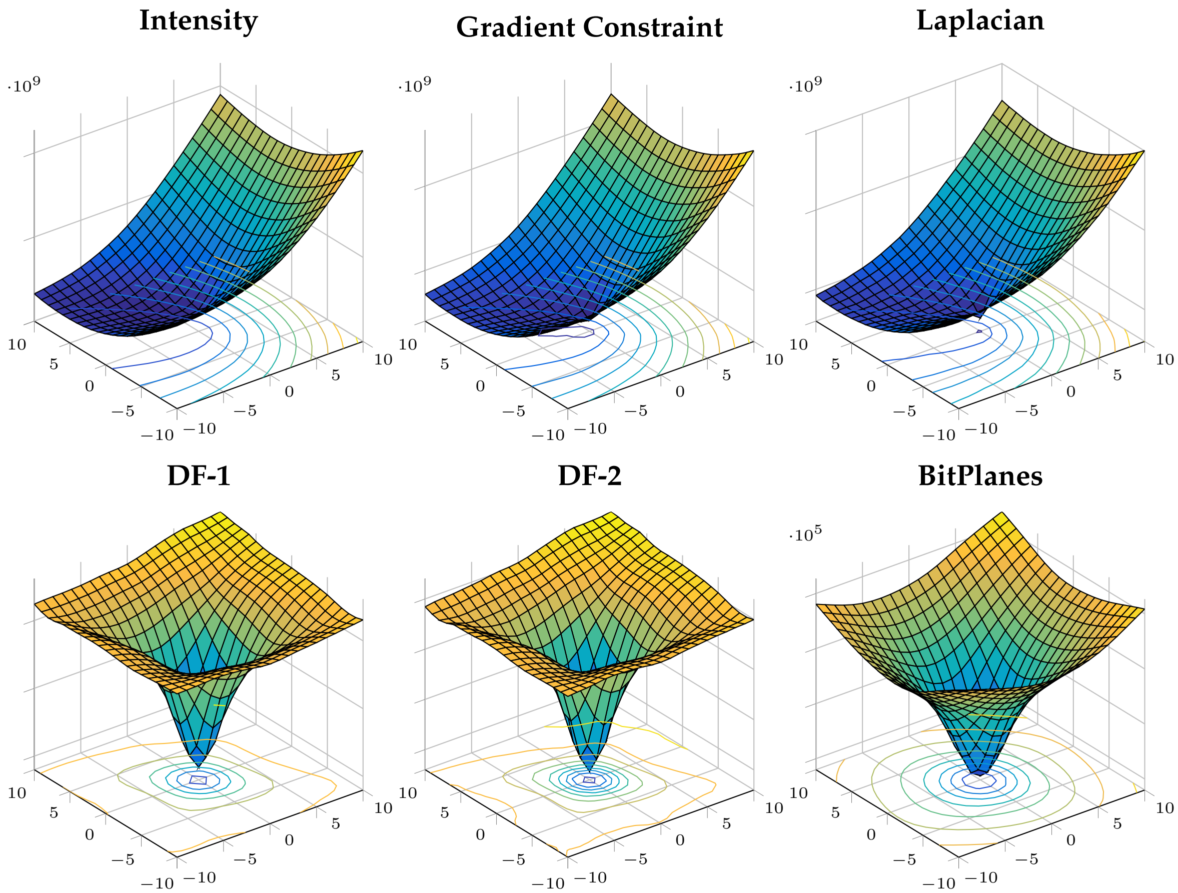


__Figure 6.5:__ Cost surfaces for each of the descriptors corresponding to the input pair shown in Fig. 6.4. The correct minima located at $(0, 0)$. Raw intensity and the gradient constraint fail to capture the correct minima. The Laplacian correctly localizes the minima, albeit a narrow basin of convergence. Descriptors at the bottom row correctly identify the minima with an adequate basin of convergence.

the nonlinear warping function depends on possibly sparse depth.

6.5.1 Affine Template Alignment

Using the notation introduced in Section 6.3, we desire to estimate the parameters of motion between a dense descriptor designated as the template $\mathbf{\Phi}_{\text{DESC}}$ and a dense descriptor evaluated on an input image $\mathbf{\Phi}'_{\text{DESC}}$, where DESC is one of the descriptors in Table 6.1. Under affine motion $\boldsymbol{\theta} \in \mathbb{R}^6$, the image coordinates of the two descriptors are related via an affine warp of the form

$$\mathbf{p}' \equiv \mathbf{w}(\mathbf{p}; \boldsymbol{\theta}) = \mathbf{A}(\boldsymbol{\theta})\mathbf{p}, \tag{6.9}$$

where $\mathbf{A}(\boldsymbol{\theta}) \in \mathbb{R}^{2\times 3}$ represents a 2D affine transform. Namely,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1+\theta_1 & \theta_3 & \theta_5 \\ \theta_2 & 1+\theta_4 & \theta_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{6.10}$$

Performance under ideal conditions

While ideal imaging conditions are challenging outside of the laboratory and controlled imaging applications (such as factory inspection), it is important to study the effect of any form of image deformation on the system's accuracy. Ideal conditions in this context are understood to mean a satisfied brightness constancy without appearance variations between the template and input images.

The question we answer in the following experiments is: *How does nonlinear deformations of the image (feature descriptors) affect estimation accuracy under ideal conditions?* Especially under the additional quantization effects caused by descriptors. The answer to question is shown in Fig. 6.6, where all descriptors are evaluated without additional illumination change. We added randomly generated Gaussian noise to obtain meaningful statistics.

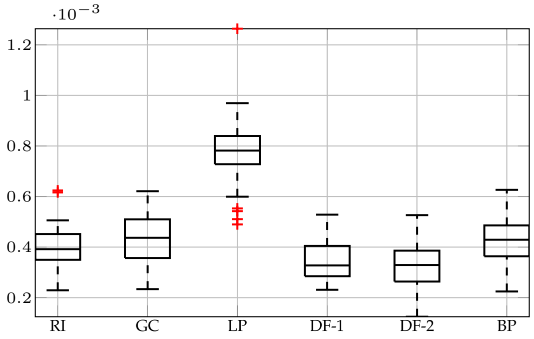


Figure 6.6: *Performance of dense descriptor under ideal conditions. Except for the Laplacian, there appears to be no significant loss of precision in comparison to working with image data directly (raw intensity values).*

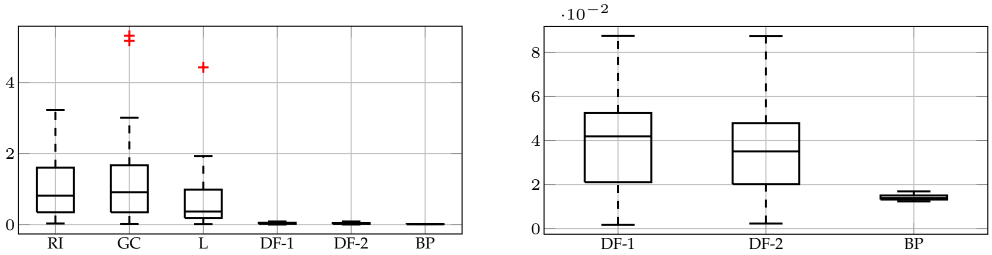


Figure 6.7: *Accuracy under illumination change. On the left RMSE is shown for all compared descriptors. On the right, we show only the top three for better comparison.*

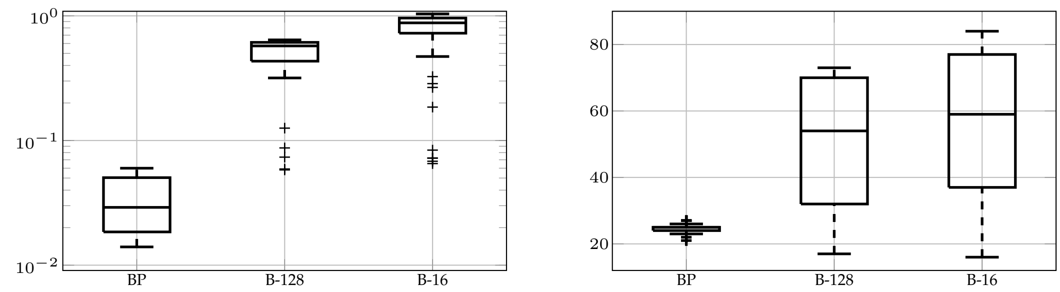
6.5.2 Performance under varying illumination

To generate illumination variations we synthesise the input image from the template using a nonlinear intensity change model of the form

$$\mathbf{I}'(\mathbf{p}) = \texttt{floor}\left(255\left(\frac{\alpha\mathbf{I}(\mathbf{w}(\mathbf{p};\boldsymbol{\theta}))) + \beta}{255}\right)^{1+\gamma}\right), \tag{6.11}$$

where $\boldsymbol{\theta}$ is a randomly generated vector of warp parameters, α and β are respectively multiplicative and additive terms, while $|\gamma| < 1$ adds a nonlinear gamma correction term. An example of this type of illumination change is illustrated in Fig. 6.4.

Results are shown in Fig. 6.7 using the end-point RMSE metric [25]. A expected, we observe a large RMSE when using raw intensities. No significant improvement is obtained using the gradient constraint. The Laplacian improves results only slightly. The top performing algorithms are DF, and Bit-Planes. As we shall see later, however, Bit-Planes' performance is more robust under more challenging illumination variations.



(a) *Error*

(b) *Number of iterations*

Figure 6.8: *Comparison with BRIEF using 128 channels (B-128) and 16 channels (B-16).*

Table 6.2: *Planar template tracking runtime until convergence in frames per second (FPS) on a single core Intel i7-2460M @ 2.8 Ghz.*

	Template area			
	75×57	150×115	300×230	640×460
Intensity	650	360	140	45
Bit-planes	460	170	90	35

6.5.3 Nonlocal descriptors

Another natural question to ask is whether there is any benefit from using nonlocal feature descriptors in direct tracking? By nonlocal, we mean feature descriptors that make use of nonlocal spatial information in the image, such as a making use of a large neighborhood during the descriptor's computation. Examples include SIFT [234], HOG [89], and BRIEF [63]. As shown in Fig. 6.8, the use of nonlocal descriptor appears to hurt performance rather than simpler local ones. In this experiment, we experiment with two possibilities of extracting channels from the BREIF [63] descriptor. One, is extracting 128 channels similar to [14]. The other, is extracting only 16 channels, where each channel is formed of a single byte. We observed similar degradation in performance using densely evaluated SIFT, and other variations on extracting channels.

6.6 Robust Planar Template Tracking

We evaluate the performance of bit-planes for a template tracking problems using the benchmark dataset collected by Gauglitz *et al.* [136]. An example of the dataset is shown in

Fig. 6.10. Our plane tracker estimates an 8DOF homography using the IC algorithm [25]. The template is extracted from the first frame in each sequence and is kept fixed throughout as we are interested in tracking robustness overtime. To improve convergence we use a 3-level pyramid and initialize the tracker for subsequent frames using the most recent estimate. We use Gauss-Newton as the optimization algorithm, without robust weighting, and with a maximum of 100 iterations. Tracking terminates early if the relative change in the estimated parameters drops below 1×10^{-6}, or the relative change in the cost function drops below 1×10^{-5}. For small motions, the tracker typically converges in less than 10 iterations using bit-planes, or raw intensities. Our implementation runs faster than real time as shown in Table 6.2. The efficiency is achieved by utilizing SIMD instructions on the CPU, which allow us to process 16 pixels at once (or 32 pixels with AVX instructions). Additionally, the operations required to compute the descriptor are limited to bit shifts, ORs and ANDs, all of which can be performed with high throughput and low latency.

We compare the performance of our algorithm against a variety of template tracking methods summarized in Table 6.3. The algorithms are: the enhanced correlation coefficient **ECC** [112], which serves as an example of an intrinsically robust cost function that is invariant up to an affine illumination change. The Dual Inverse Compositional (DIC) algorithm [36], which severs as an example of algorithms that attempt to estimate illumination parameters. We use two variations of the DIC: (i) the gain+bias model on grayscale images denoted by **DIC-1**, and (ii) using a full affine lighting model the makes use of RGB image data denoted by **DIC-2**. We also compare the performance against a recently published descriptor-based method [83] called Descriptor Fields **DF**. Finally, we include baseline results from raw intensity **LK**, improved LK with the Gradient Constraint **GC** [61], and alignment with the Gradient Magnitude **GM**.

We report two quantities in the evaluation. First, is the percentage of successfully tracked frames. A frame is successfully tracked if the overlap between the estimate and the ground truth is greater than 90%. The overlap is computed as $o = (A \cap B)/(A \cup B)$, where A is the warped image given each algorithm's estimate, and B is the warped image given the ground truth. Second, since we are also interested in subpixel accuracy we show the mean percentage of overlap across all frames given by $m = {}^{1}\!/\!{}_{n} \sum_{i=1}^{n} o_i$, where n is the number of frames in each sequence.

Results are compared for three types of geometric and photometric variations. First is an **out of plane rotation**, which induces perspective change as shown in Fig. 6.10b. Second, is **dynamic lighting change** where the image is stationary but illuminated with nonlinearly varying light source. Finally, a **static lighting change**, where illumination change is

Table 6.3: *Algorithms compared in this chapter. The number of parameters indicates the DOF of the state vector, which is 8 for a homography in addition to any photometric parameters. We use the authors' code for ECC and DIC.*

	# parameters	# channels
BP (ours)	8	8
ECC [112]	8	1
DIC-1 [36]	10	1
DIC-2 [36]	20	3
DF [83]	8	5
GC	8	3
GM	8	2
LK	8	1

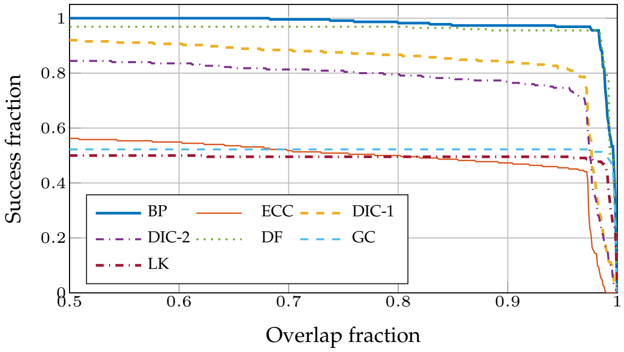


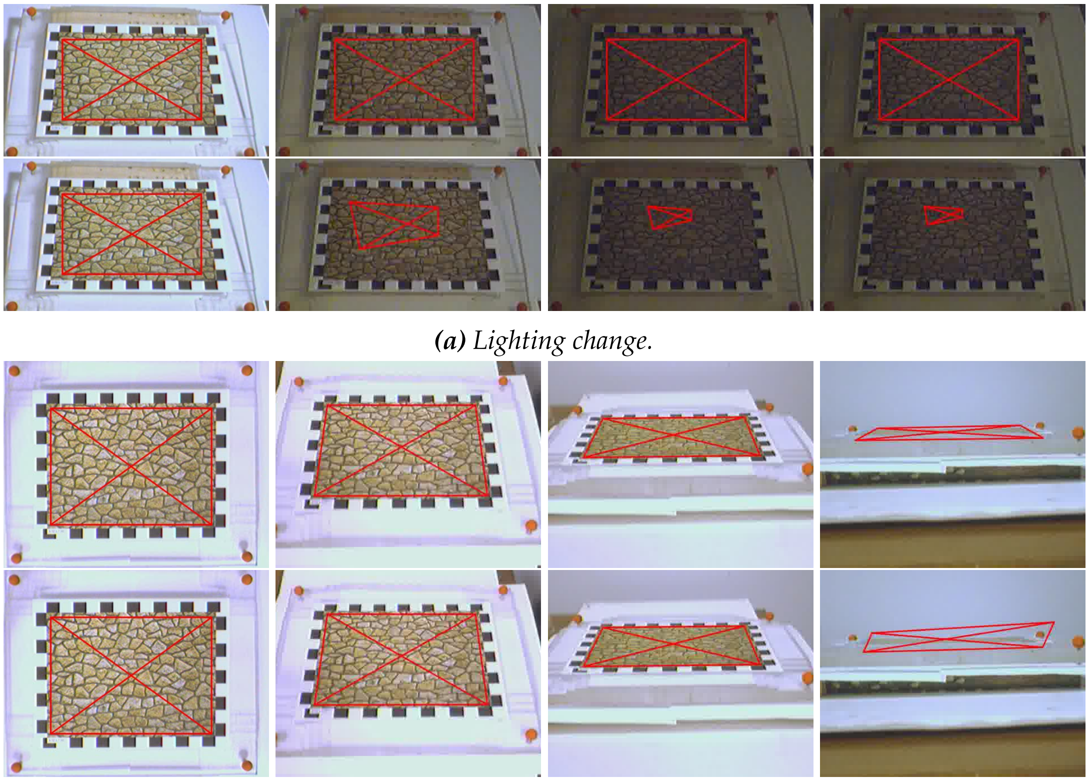
Figure 6.9: *Fraction of successfully tracked frames as function of the overlap area given the ground truth. Bit-planes and DF perform better than other methods. However, in Table 6.4 we see that bit-planes' performance is better with challenging sequences.*

sudden.

Our evaluation results are shown in Table 6.4 and in Fig. 6.9. The top performing methods are based on a descriptor constancy assumption, namely: BP and DF. However, BP is more efficient and it performed significantly better for the out of plane rotation data. In fact, all tested algorithms, except BP, performed poorly with this data. Algorithms that use a robust function (ECC) and the ones that attempt to estimate illumination (DIC) performed well, but fell behind in comparison to descriptor constancy and even the gradient constraint.

Table 6.4: *Template tracking evaluation [136]. We show the percentage of successfully tracked frames. In parentheses we show the average percentage of overlap for all successfully tracked frames. The available textures are:* `br` *(bricks),* `bu` *(building),* `mi` *(mission),* `pa` *(paris),* `su` *(sunset), and* `wd` *(wood).*

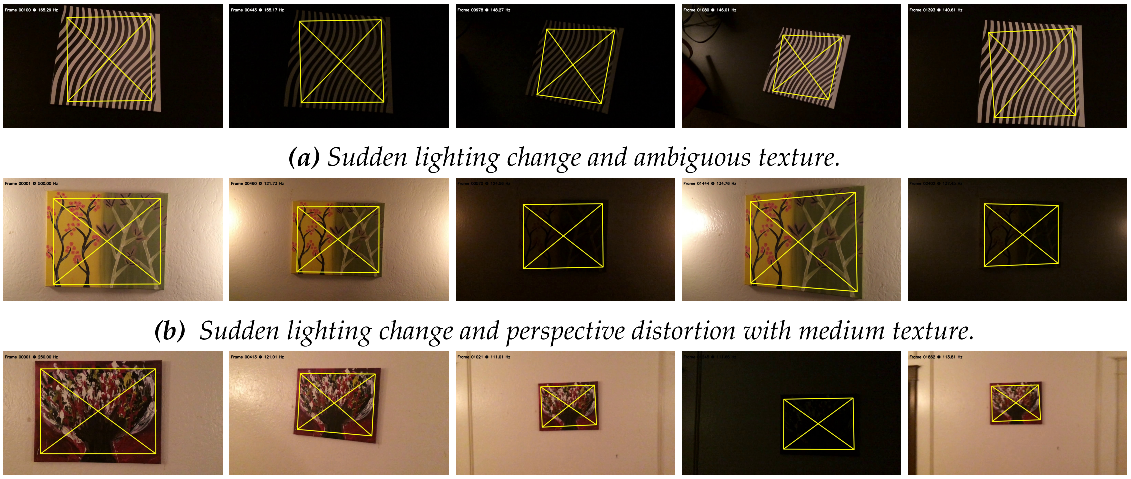
	`br`	`bu`	`mi`	`pa`	`su`	`wd`
	Out of Plane Rotation					
BP	**100.0** (99.38)	**100.0** (99.51)	**87.50** (99.38)	**97.92** (99.26)	**79.17** (99.57)	**93.75** (99.30)
ECC	25.00 (96.16)	33.33 (95.85)	25.00 (95.99)	33.33 (96.65)	20.83 (95.52)	18.75 (95.14)
DIC-1	25.00 (96.20)	33.33 (95.83)	25.00 (95.98)	33.33 (96.73)	20.83 (95.95)	18.75 (95.46)
DIC-2	25.00 (96.22)	35.42 (95.56)	25.00 (95.51)	35.42 (96.42)	25.00 (96.22)	18.75 (95.06)
DF	91.67 (99.51)	93.75 (99.44)	79.17 (99.70)	85.42 (99.75)	70.83 (99.60)	83.33 (99.51)
GC	**100.0** (99.24)	95.83 (99.66)	87.50 (99.52)	93.75 (99.51)	62.50 (98.88)	91.67 (99.34)
GM	62.50 (99.86)	83.33 (99.62)	77.08 (99.72)	77.08 (99.81)	58.33 (99.71)	62.50 (99.66)
LK	93.75 (99.68)	91.67 (99.70)	83.33 (99.32)	91.67 (99.63)	37.50 (97.64)	66.67 (99.63)
	Dynamic Lighting Change					
BP	**100.0** (98.97)	**100.0** (99.08)	**100.0** (99.13)	**100.0** (98.91)	**100.0** (98.98)	**100.0** (99.02)
ECC	16.33 (98.03)	19.39 (99.00)	**100.0** (98.64)	**100.0** (98.69)	**100.0** (97.30)	67.35 (98.55)
DIC-1	**100.0** (98.40)	**100.0** (99.04)	**100.0** (98.77)	**100.0** (98.60)	86.87 (96.02)	20.41 (95.36)
DIC-2	**100.0** (98.39)	**100.0** (98.85)	**100.0** (98.61)	**100.0** (98.58)	85.86 (96.42)	26.53 (97.73)
DF	**100.0** (99.30)	**100.0** (99.08)	**100.0** (98.35)	**100.0** (98.87)	20.41 (99.36)	68.37 (99.02)
GC	17.35 (99.87)	**100.0** (99.50)	22.45 (99.84)	18.37 (99.88)	12.24 (99.72)	17.35 (99.84)
GM	17.35 (98.99)	19.39 (99.23)	23.47 (99.10)	19.39 (99.08)	0.00 (0.00)	0.00 (0.00)
LK	13.27 (99.34)	31.63 (98.26)	18.37 (98.82)	18.37 (99.32)	12.24 (99.16)	16.33 (98.96)
	Static lighting change					
BP	**100.0** (99.76)	**100.0** (99.85)	**100.0** (99.61)	**100.0** (99.85)	**100.0** (99.63)	**100.0** (99.76)
ECC	**100.0** (97.33)	**100.0** (97.67)	**100.0** (97.75)	**100.0** (97.41)	**100.0** (96.79)	**100.0** (97.55)
DIC-1	**100.0** (97.70)	**100.0** (97.77)	**100.0** (97.80)	**100.0** (97.20)	98.72 (96.58)	89.74 (96.19)
DIC-2	**100.0** (97.58)	79.49 (97.59)	**100.0** (97.07)	**100.0** (97.13)	89.74 (95.75)	79.49 (96.38)
DF	**100.0** (99.68)	**100.0** (99.51)	76.92 (99.71)	**100.0** (99.77)	74.36 (99.70)	**100.0** (99.83)
GC	74.36 (99.73)	74.36 (99.84)	48.72 (99.97)	74.36 (99.76)	48.72 (99.74)	51.28 (99.88)
GM	48.72 (99.88)	74.36 (99.75)	74.36 (99.66)	74.36 (99.81)	48.72 (99.76)	48.72 (99.83)
LK	48.72 (99.80)	74.36 (99.67)	48.72 (99.95)	48.72 (99.93)	48.72 (99.40)	48.72 (99.94)



(a) *Lighting change.*

(b) *Out-of-plane rotation.*

Figure 6.10: *Tracking results using the Bricks dataset [136]. The top row of each figure shows the performance of bit-planes, while the bottom row shows classical intensity-based LK.*



(a) *Sudden lighting change and ambiguous texture.*

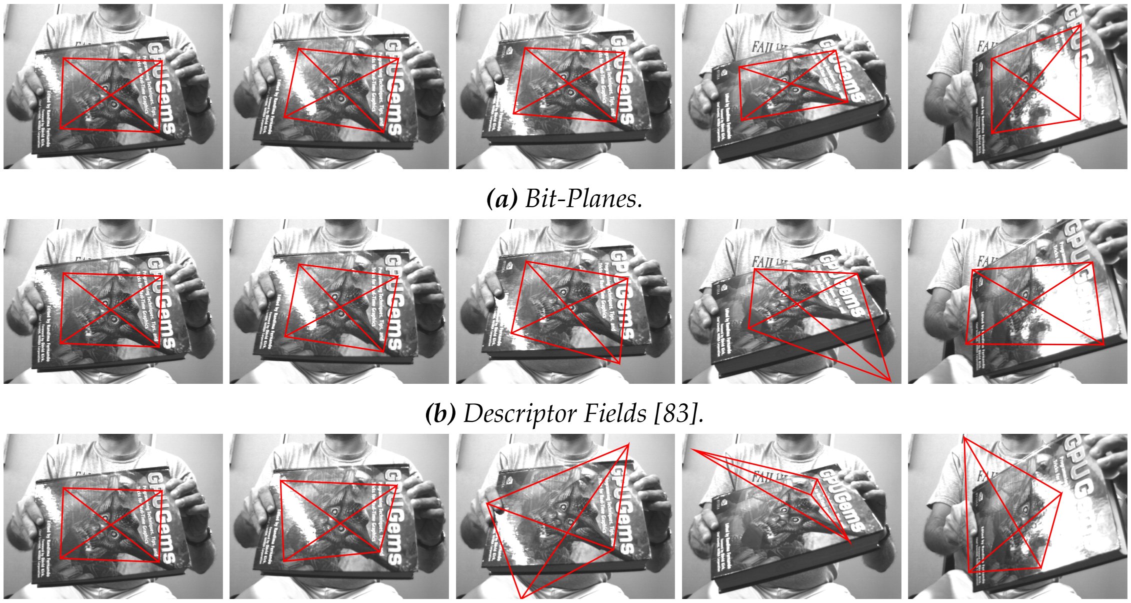
(b) *Sudden lighting change and perspective distortion with medium texture.*

(c) *Sudden lighting change and motion blur with high texture.*

Figure 6.11: *High frame rate data at* 120 *Hz captured using an iPhone 5s. Dataset contains different textures under sudden lighting change, low lighting, and motion blur. Data and code are available on https://www.cs.cmu.edu/~halismai/bitplanes. Additional results demonstraing robustness to specular reflections are shown in Fig. 6.12*

Table 6.5: *Template tracking running time on ARM architecture using a single CPU core in frames per second (FPS). The bottleneck for bit-planes is image resizing and warping, which could be alleviated using the GPU. Results are averaged over three videos of challenging data totalling 6446 frames.*

	iPad Air 2			iPhone 5s		
template size	BP	ORB	BRISK	BP	ORB	BRISK
70 × 55	**123**	N/A	N/A	**50**	N/A	N/A
150 × 115	**48**	15	15	**22**	13	13
311 × 230	**17**	12	14	**10**	8	11



(a) *Bit-Planes.*

(b) *Descriptor Fields [83].*

(c) *Gradient Constraint*

Figure 6.12: *Illustration of robustness to specular reflections in comparision to other tracking algorithms using the "book" dataset [334]. Video demonstration is available on* `https://goo.gl/sFoVRP`

6.6.1 Results on mobile devices

We further evaluate the work on high frame rate data (Slo-mo) using two smart mobile devices: the iPad Air 2 and the iPhone 5s. In addition to compression artifacts, we made the data more challenging by turning off the lights multiples times during acquisition to cause sudden lighting change and low illumination. The videos are recorded with unsteady hands causing further motion blur. An example of the videos is shown in Fig. 6.11 featuring an ambiguously textured object in Fig. 6.11a, normal levels of texture in Fig. 6.11b as well as higher amount of texture in Fig. 6.11c. The first image in Fig. 6.11 shows the selected template, which we hold fixed throughout tracking. The total number of frames from the videos combined is 6447.

We compare the performance of dense tracking using bit-planes with the RANSAC-based track by detection using two types of binary descriptors, ORB [313] and BRISK [221]. In terms of efficiency, even though our mobile device implementation does not make use of NEON instructions or the GPU, we outperform OpenCV3's optimized implementations of ORB and BRISK by a substantial margin. More importantly, our approach is more robust. Feature-based tracking failed on $\approx 15\%$ of the frames due to either the inability to detect features under low light, or failure due to imprecise correspondences under motion blur.

Perhaps more interestingly, bit-planes is able to maintain performance with smaller image resolution. In fact, tracking speed more than doubles when reducing the template size by half. However, this is not the case with sparse features as memory overhead depends on the number of extracted keypoints, which we kept fixed at 512. It is possible to improve the tracking speed of ORB and BRISK by reducing the number of extracted keypoints. However, lowering the number of keypoints must be done carefully as not to compromise the robustness of the system. We note that the ability to work with lower resolution is important on mobile devices to lower power consumption.

Finally, we note that while dense bit-planes tracking produces faster and more accurate results, its main limitation is the inability to recover if the template is lost due to occlusions or significant drift. In such cases, track by detection can be of immense value to re-initialize LK-based methods if needed.

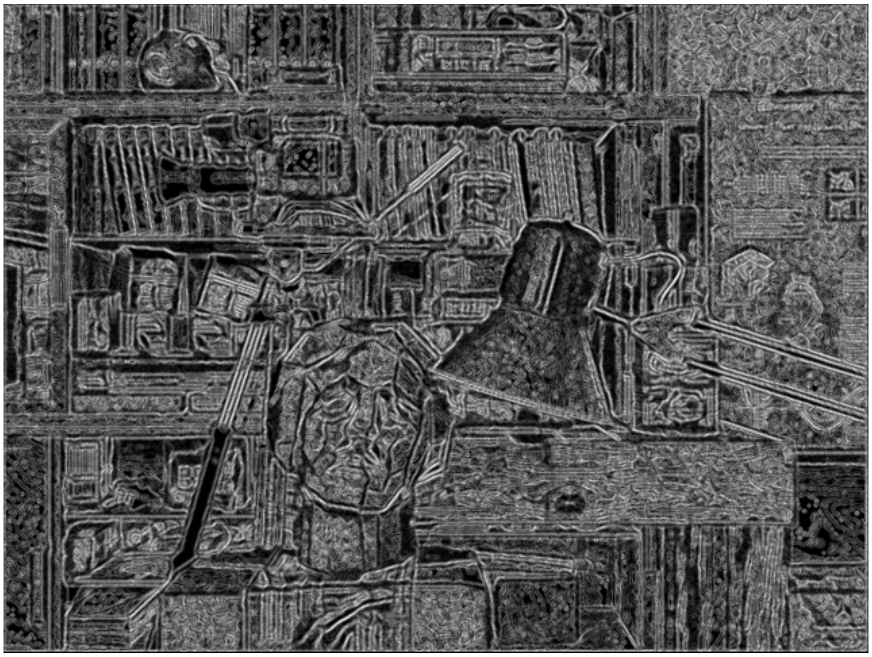


Figure 6.13: *The absolute gradient magnitude of Bit-Planes over all channels, which we use for pixel selection. Darker values are smaller.*

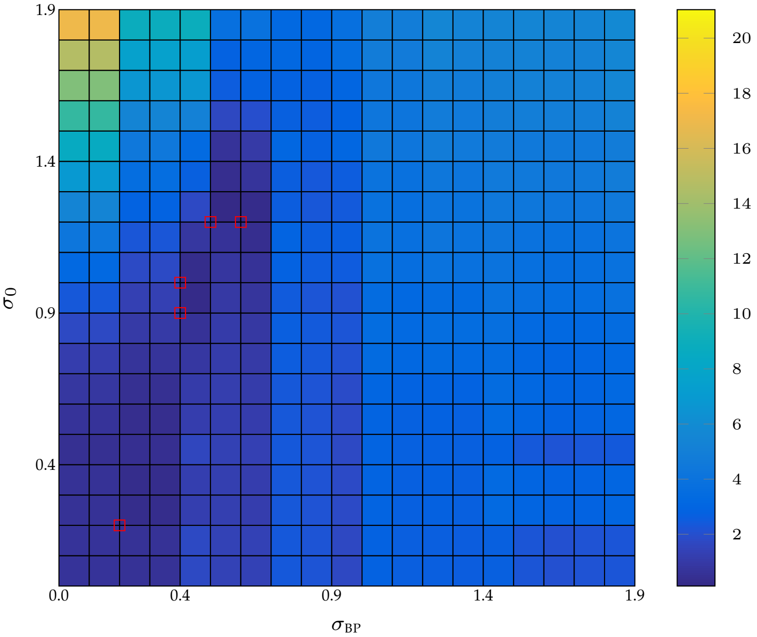


Figure 6.14: *Error as function of pre-smoothing the image with a Gaussian kernel of standard deviation of* σ_0 *as well as smoothing the Bit-Planes with* σ_1*. The lowest error is associated with smaller kernels.*

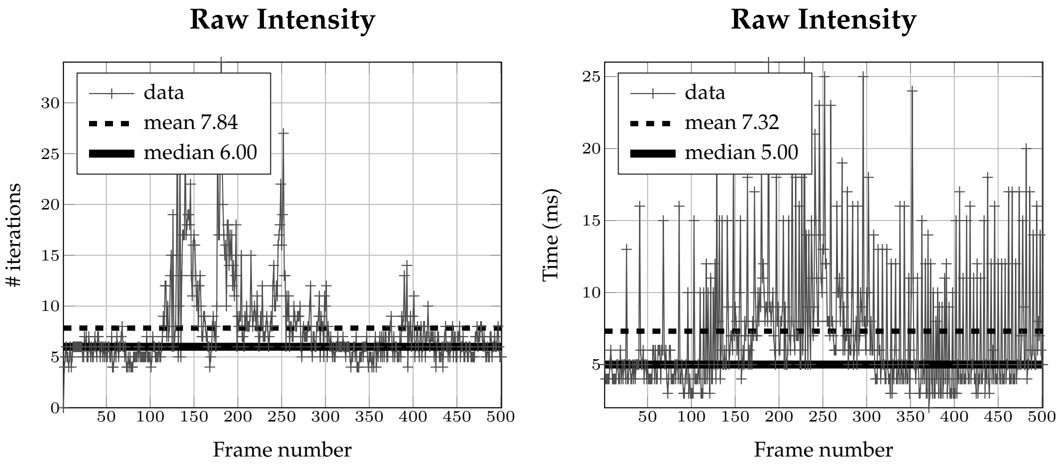


Figure 6.15: *Number of iterations and runtime on the first* 500 *frames of the New Tsukuba dataset using raw intensity only. On average, the algorithm runs at more than* 100 *Hz.*

6.7 Robust Visual Odometry

6.7.1 Effect of smoothing

Fig. 6.14 shows the effect of smoothing the image prior to computing Bit-Planes. The experiment is performed on synthetic data with a small translational shift. Higher smoothing kernels tend to washout the image details required to estimate small motions. Hence, we use a 3×3 kernel with $\sigma = 0.5$.

6.7.2 Experiments with synthetic data

We use the "New Tsukuba" dataset [247, 299] to compare the performance of our algorithm against two representative algorithms from the state-of-the-art. The first is FOVIS [176], which we use as a representative of feature-based methods. The second is DVO [198] as representative of direct methods using the brightness constancy assumption. The most challenging illumination condition provided by the Tsukuba dataset is when the scene is lit by "lamps" as shown in Fig. 6.19, which we use in our evaluation.

Our goal is this experiment is to assess the utility of our proposed descriptor in handling the arbitrary change in illumination visible in all frames of the dataset. Hence, we initialize all algorithms with the ground truth disparity/depth map. In this manner, any pose estimation errors are caused by failures to extract and match features, or failure in minimizing the photometric error. As shown in Fig. 6.17 and Fig. 6.18 the robustness of our approach

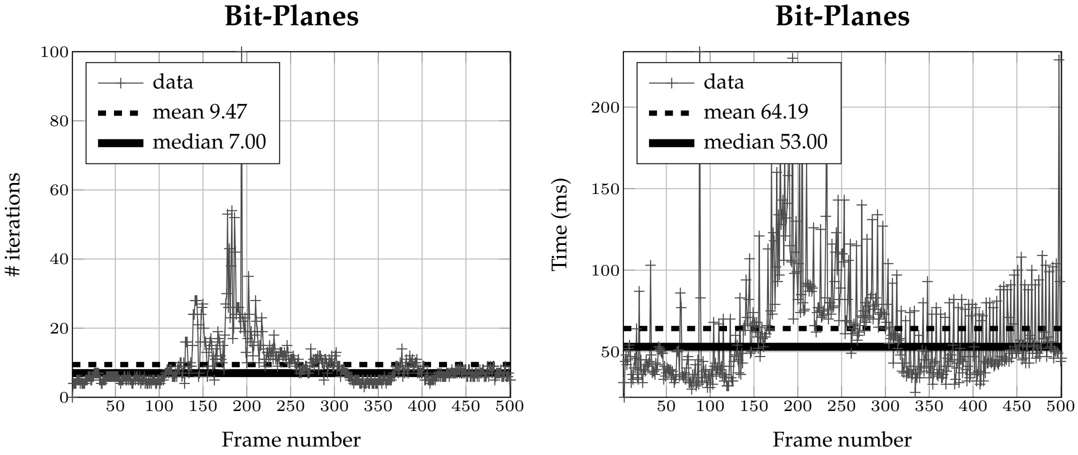


Figure 6.16: Number of iterations and run time using on the first 500 *frames of the New Tsukuba dataset using the Bit-Planes descriptors. On average, the algorithm runs at* 15 *Hz.*

far exceeds the conventional state-of-the-art. Also, as expected, feature-based methods in this case (FOVIS) slightly outperforms direct methods (DVO) due to the challenging illumination of the scene.

6.7.3 Evaluation on the KITTI benchmark

The KITTI benchmark [138] presents a challenging dataset for our algorithm, and all direct methods in general, as the motion between consecutive frames is large. The effect of large motions can be observed in Fig. 6.22, where the performance of our algorithm noticeably degrades at higher vehicle speeds. This limitation could be mitigated by using a higher camera frame rate, or providing a suitable initialization.

6.7.4 Real data from underground mines

We demonstrate the robustness of our algorithm using data collected in underground mines. Our robot is equipped with a stereo camera with 7cm baseline that outputs grayscale images of size 1024×544 and computes an estimate of disparity using a hardware implementation of SGM [168]. An example VO result along with a sample of the data is shown in Fig. 6.23.

Due to lack of lighting in underground mines, the robot carries its own source of LED light. However, the LEDs are insufficient to uniformly illuminate the scene due to power constraints in the system. We have attempted to use other open source VSLAM/VO pack-

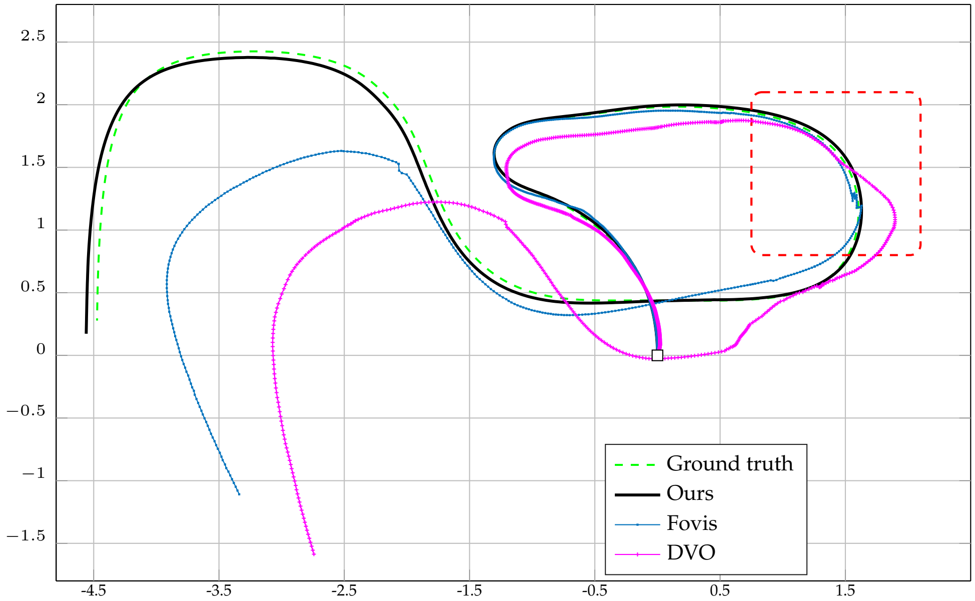


Figure 6.17: Evaluation on the synthetic Tsukuba sequence [299] using the illumination provided by "lamps" in comparison to other VO algorithms. The figure shows a bird's eye view of the estimated trajectory of the camera from each algorithm in comparison to the ground truth. The highlighted area is shown with more details in Fig. 6.18. Example images are in Fig. 6.19.

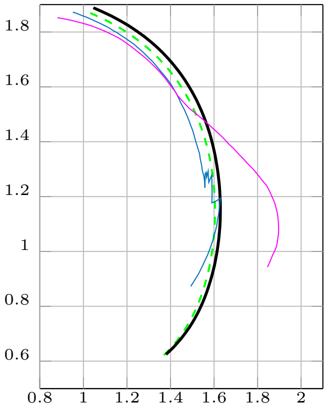


Figure 6.18: Estimated camera path details for each of the algorithms shown in Fig. 6.17.

.

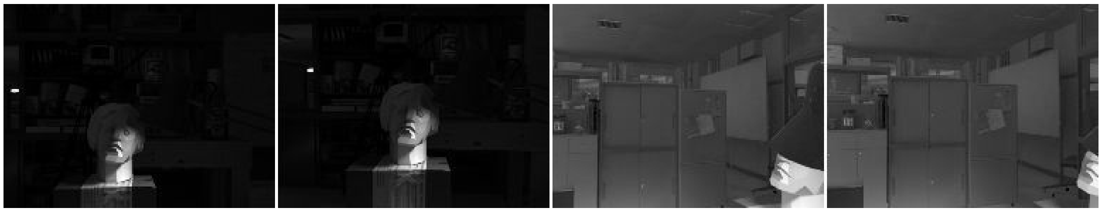


Figure 6.19: Example images from the "lamps" sequence.

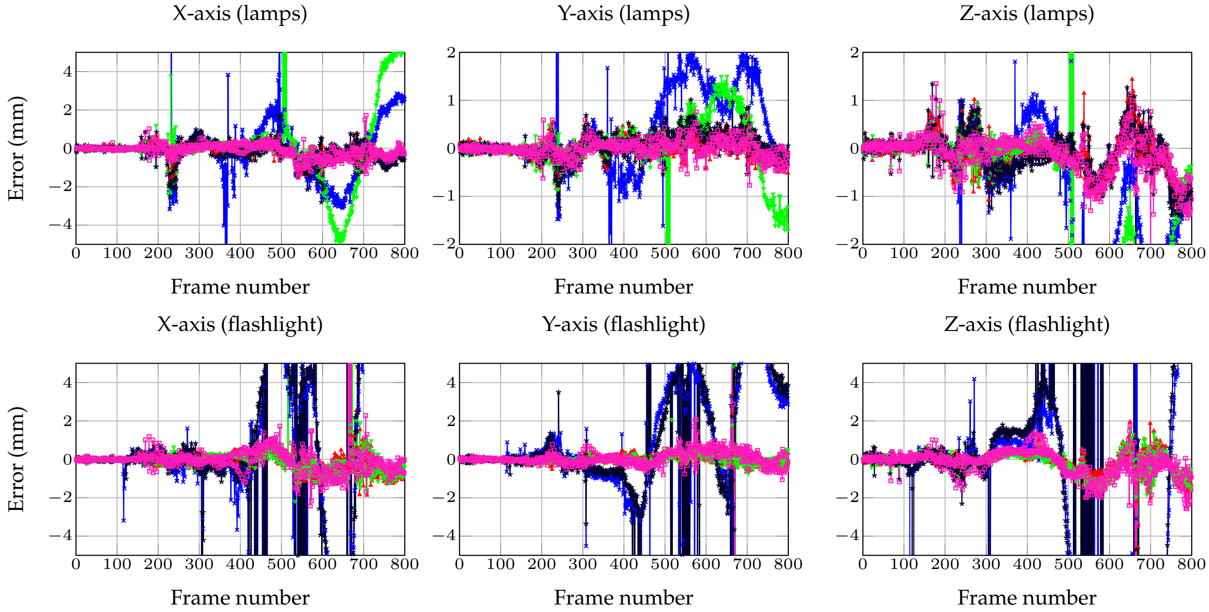


Figure 6.20: *Trajectory errors using the "lamps" sequence for (RI —×—), (LP —★—), (DF-1 —▲—), (DF-2 —•—), and (BP —□—). We truncated the plots for better visualization.*

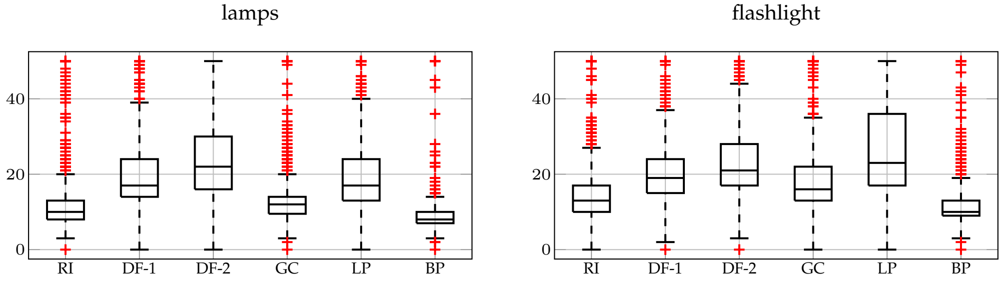


Figure 6.21: *Number of iterations.*

Table 6.6: *Summary statistics of errors per positional degree of freedom (RMSE in mm). We use the standard right-handed coordinate convention system in vision, where the Z-axis points forward and the Y-axis points downward.*

	flahslight			lamps		
	X	Y	Z	X	Y	Z
RI	14.34	8.14	20.94	1.45	1.01	2.16
GC	14.26	7.53	18.86	45.31	30.37	22.76
LP	13.24	6.59	18.16	0.54	0.26	0.46
DF-1	**2.03**	0.45	**0.77**	0.42	0.21	**0.40**
DF-2	2.95	0.83	1.33	2.27	1.09	4.90
BP	2.66	**0.33**	1.08	**0.37**	**0.18**	**0.40**

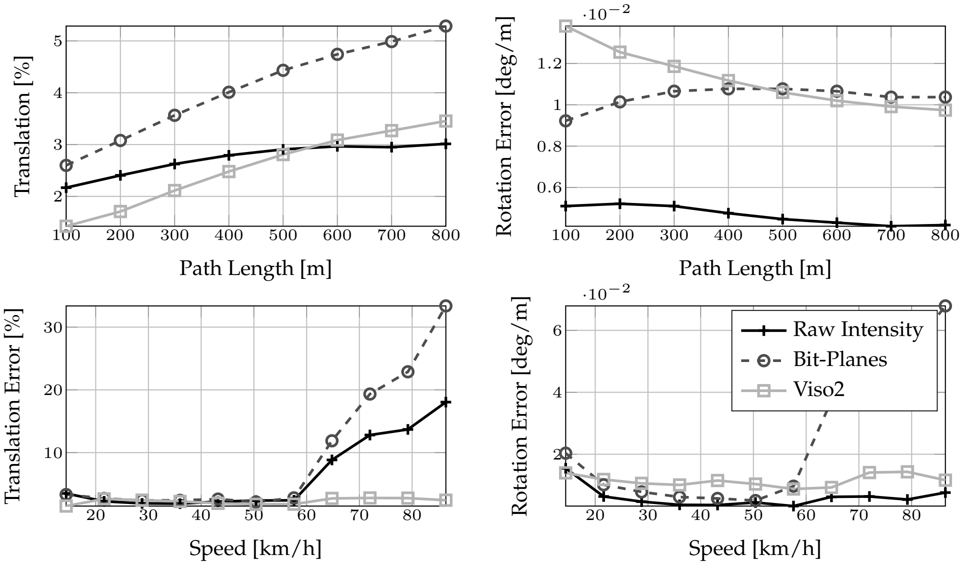
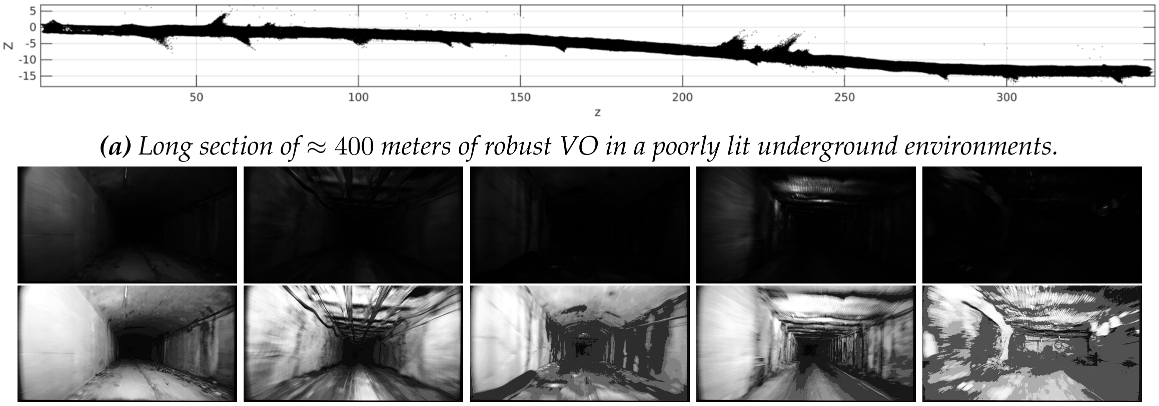


Figure 6.22: *Performance on the training data of the KITTI benchmark in comparison to VISO2 [137]. The large baseline between consecutive frames presents a challenge to direct methods as can be seen by observing the error as a function of speed. Nonetheless, rotation accuracy of our method remains high.*



(a) *Long section of* ≈ 400 *meters of robust VO in a poorly lit underground environments.*

Figure 6.23: *Example result and representative images from the first mine sequence (top row) and a histogram equalized version for visualization (bottom row).*

ages [137, 176, 276], but they all fail too often due to the severely degraded illumination conditions.

In Fig. 6.24, we show another result from a different underground environment where the stereo 3D points are colorized by height. The large empty areas in the generated map is due to lack of disparity estimates in large portions of the input images. Due to lack of ground-truth we are unable to assess the accuracy of the system. But, visual inspection of the created 3D maps indicate minimal drift, which is expected when operating in an open loop fashion.

6.7.5 Reconstruction density

Density of the reconstructed point cloud is demonstrated in Fig. 6.25 and Fig. 6.26. Denser output is possible by eliminating the pixel selection step at the expense of increased computational time.

6.7.6 Failure cases

Most failure cases are due to a complete image washout. An example is shown in Fig. 6.27. Theses cases occur when the robot is navigating a tight turn such that all the LED output is constrained very closely to the camera. Addressing such cases, form vision-only data, is a good avenue of future work.

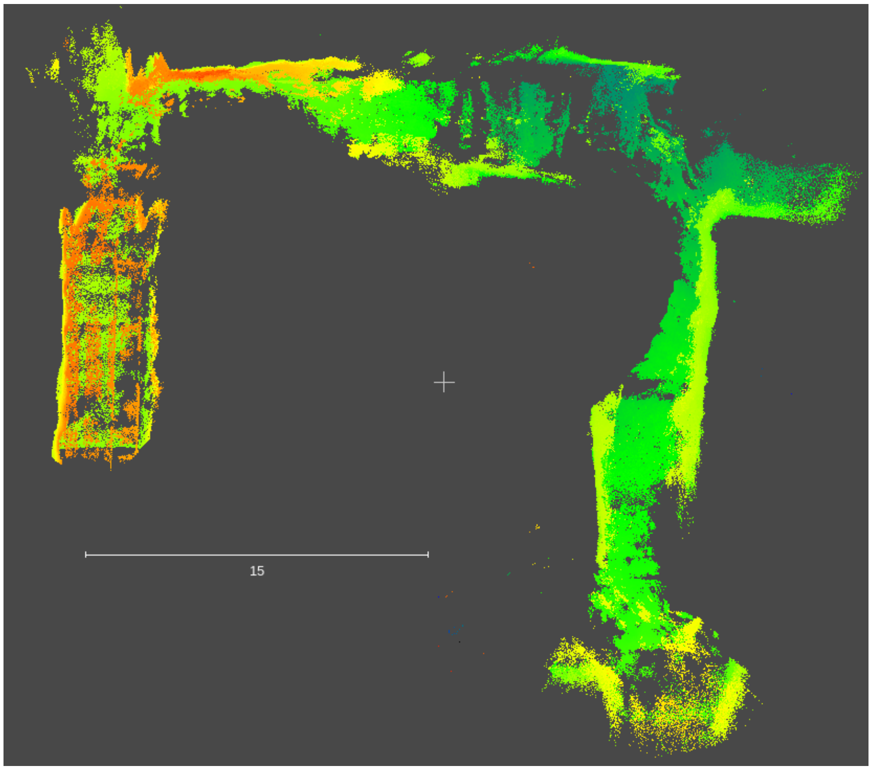


Figure 6.24: *VO map colorized by height showing the robot transitioning between different levels in the second mine dataset.*

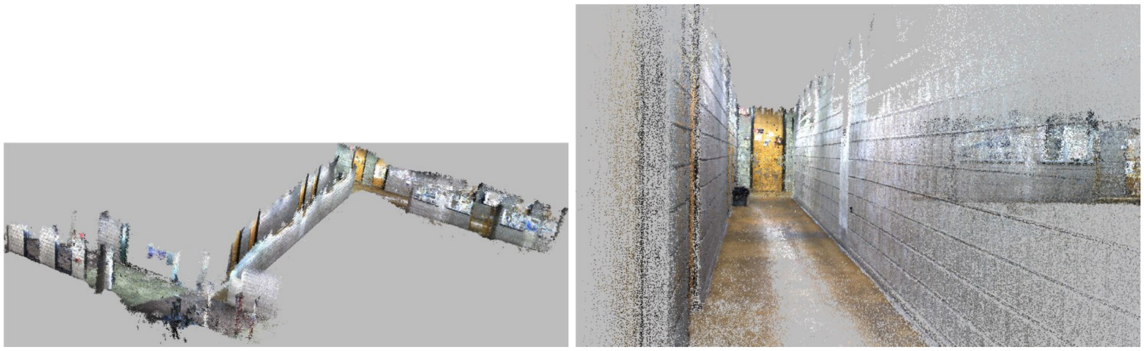


Figure 6.25: *Reconstruction density using the Weal hall dataset Alismail et al. [11].*

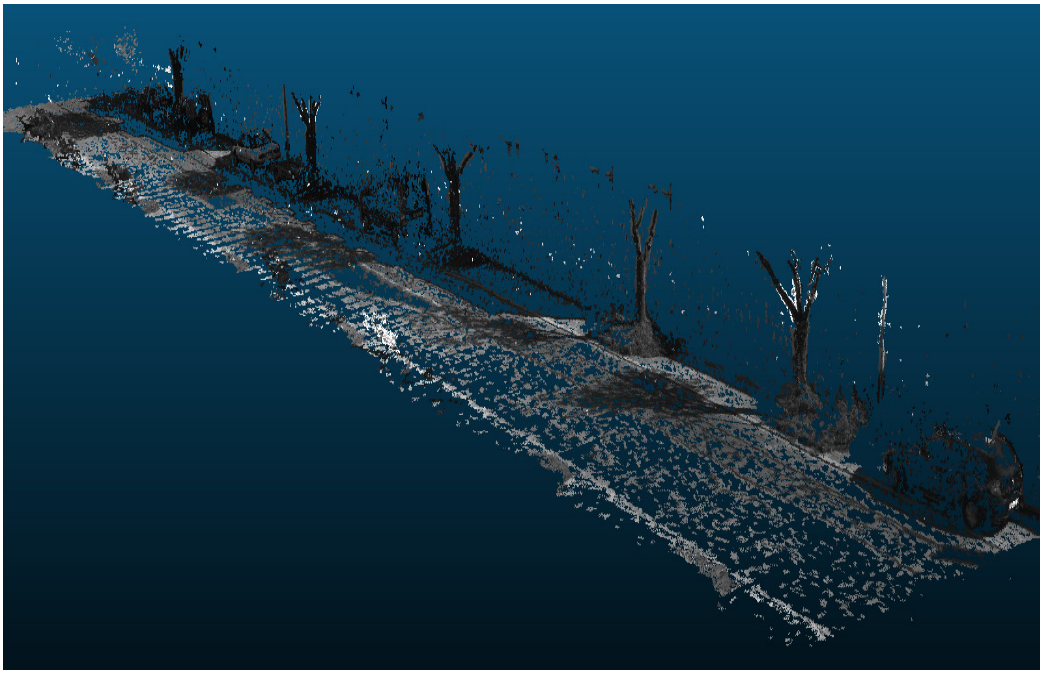


Figure 6.26: Reconstruction density on a section of the KITTI dataset.

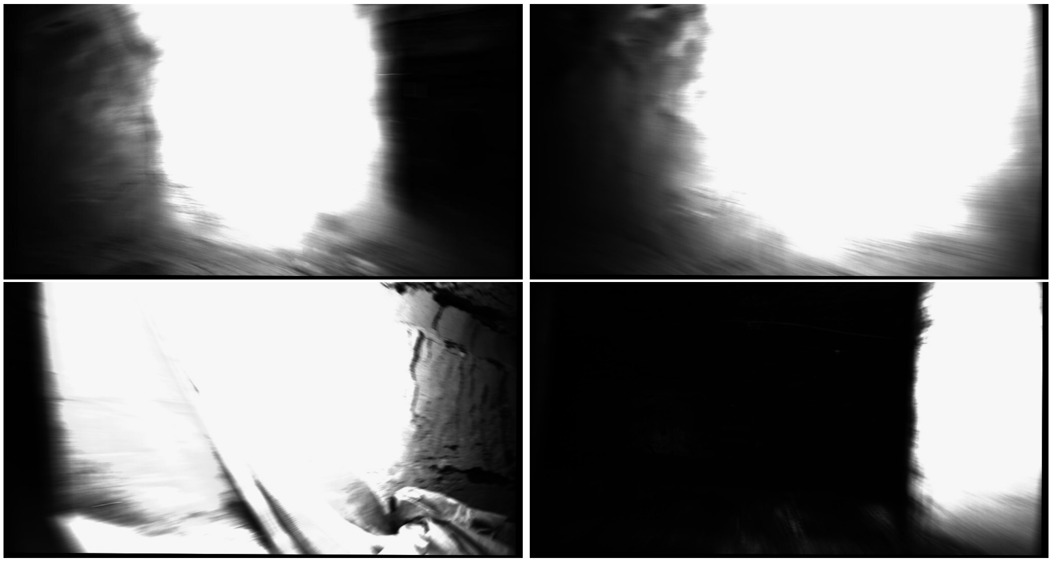


Figure 6.27: An example of a failure case where most of the image details are washed out causing complete loss of stereo.

6.8 Summary

In this work, we presented a VO system capable of operating in challenging environments where the illumination of the scene is poor and non-uniform. The approach is based on direct alignment of feature descriptors. In particular, we designed an efficient to compute binary descriptor that is invariant to monotonic changes in intensity. By using this descriptor constancy, we allow vision-only pose estimation to operate robustly in environments that lack keypoints and lack the photometric consistency required by direct methods.

Our descriptor, Bit-Planes, is designed for efficiency. However, other descriptors could be used instead (such ORB and/or SIFT) if computational demands are not an issue. A comparison of performance between difference descriptors in a direct framework is an interesting direction of future work as their amenability to linearization may differ.

The approach is simple to implement, and can be readily integrated into existing direct VSLAM algorithms with a small additional computational overhead.

CHAPTER 7

Bundle Adjustment without Correspondences

Contents

7.1 Summary of Contributions

- We develop a novel formulation for VSLAM using a direct (photometric) approach where photometric constancy is maximized across multiple views in a sliding window fashion.

- In contrast to previous work, we show that the joint refinement of motion and structure is feasible in a direct approach.

- The approach is evaluated on a range of outdoor datasets and is shown to outperform state-of-the-art methods based on the minimization of the reprojection error using bundle adjustment.

- The results obtained in this chapter indicate the minimizing the reprojection error, while in theory may be optimal there are sources of errors that are not addressed. Brining back the image in the loop appears to address some of these feature localization errors and improve on the accuracy of results.

7.2 Introduction

Photometric, or image-based, minimization is a fundamental tool in a myriad of applications such as: optical flow [352], scene flow [384], and stereo [132, 324]. Its use in vision-based 6DOF motion estimation has recently been explored demonstrating good results [110, 198, 255, 345]. Minimizing the photometric error, however, has been limited to frame–frame estimation (visual odometry), or as a tool for depth refinement disjoint from the parameters of motion [285]. Consequently, in unstructured scenes, frame–frame minimization of the photometric error cannot reduce the accumulated drift. When loop closure and prior knowledge about the motion and structure are not available, one must resort to the Gold Standard: minimizing the reprojection error using bundle adjustment.

Bundle adjustment (BA) is the problem of jointly refining the parameters of motion and structure to improve a visual reconstruction [372]. Although BA is a versatile framework, it has become a synonym to minimizing the reprojection error across multiple views [160, 369]. The advantages of minimizing the reprojection error are abundant and have been discussed at length in the literature [160, 369]. In practice, however, there are sources of systematic errors in feature localization that are hard to detect and the value of modeling their uncertainty remains unclear [58, 195]. For example, slight inaccuracies in calibration exaggerate errors [133], sensor noise and degraded frequency content of the image affect feature localization accuracy [101]. Even interpolation artifacts play a non-negligible role [330]. Although minimizing the reprojection is backed by sound theoretical properties [160], its use in practice must also take into account the challenges and nuances of precisely localizing keypoints [372].

Here, we propose a novel method that further improves upon the accuracy of minimizing the reprojection error and, even state-of-the-art loop closure [276]. The proposed algorithm brings back the image in the loop, and jointly refines the motion and structure parameters to maximize photometric consistency across multiple views. In addition to improved accuracy, the algorithm does not require correspondences. In fact, correspondences are estimated automatically as a byproduct of the proposed formulation.

The ability to perform BA without the need for precise correspondences is attractive because it can enable VSLAM applications where corner extraction is unreliable [265], as well as additional modeling capabilities that extend beyond geometric primitives [305, 315].

7.2.1 Preliminaries and Notation

The reprojection error

Given an initial estimate of the scene structure $\{\boldsymbol{\xi}_j\}_{j=1}^{N}$, the viewing parameters per camera $\{\boldsymbol{\theta}_i\}_{i=1}^{M}$, and $\mathbf{x}_{ij}$ the projection of the j^{th} point onto the i^{th} the reprojection error is given by

$$\epsilon_{ij}(\mathbf{x}_{ij}; \boldsymbol{\theta}_i, \boldsymbol{\xi}_j) = \left\| \mathbf{x}_{ij} - \pi\left(\mathbf{T}(\boldsymbol{\theta}_i), \mathbf{X}(\boldsymbol{\xi}_j)\right) \right\|, \tag{7.1}$$

where $\pi(\cdot, \cdot)$ is the image projection function. The function $\mathbf{T}(\cdot)$ maps the vectorial representation of motion to a rigid body transformation matrix. Similarly, $\mathbf{X}(\cdot)$ maps the parameterization of the point to coordinates in the scene.

In this work, we assume known camera calibration parameters as is often the case in VSLAM and parameterize the scene structure using the usual 3D Euclidean coordinates, where $\mathbf{X}(\boldsymbol{\xi}) := \boldsymbol{\xi}$, and

$$\boldsymbol{\xi}_j^\top = \begin{pmatrix} x_j & y_j & z_j \end{pmatrix} \in \mathbb{R}^3. \tag{7.2}$$

The pose parameters are represented using twists [277], where the rigid body pose is obtained using the exponential map [174], *i.e.*

$$\boldsymbol{\theta}_i^\top \in \mathbb{R}^6 \quad \text{and} \quad \mathbf{T}(\boldsymbol{\theta}) := \exp(\widehat{\boldsymbol{\theta}}) \in SE(3). \tag{7.3}$$

Our algorithm, similar to minimizing the reprojection error using BA, does not depend on the parameterization. Other representations for motion and structure have been studied in the literature and could be used as well [70, 162, 405].

Geometric bundle adjustment

Given an initialization of the scene points and motion parameters, we may obtain a refined estimate by minimizing the squared reprojection error in Eq. (7.1) across tracked features, *i.e.*

$$\left\{\Delta\boldsymbol{\theta}_i^*, \Delta\boldsymbol{\xi}_j^*\right\} = \operatorname*{argmin}_{\boldsymbol{\theta}_i, \boldsymbol{\xi}_j} \sum_{i=1}^{M}\sum_{j=1}^{N} \frac{1}{2}\delta_{ij}\epsilon_{ij}^2(\mathbf{x}_{ij}, \Delta\boldsymbol{\theta}_i, \Delta\boldsymbol{\xi}_j), \tag{7.4}$$

where $\delta_{ij} = 1$ if the j^{th} point is visible, or tracked, in the i^{th} camera. We call this formulation *geometric* BA.

Minimizing the reprojection error in Eq. (7.4) is a large nonlinear optimization problem. Particular to BA is the sparsity pattern of its linearized form, which we can exploit for both large–, and medium–scale problems [160]. A more detailed review is also available in Section 2.8 of this document.

7.2.2 The Photometric Error

The use of photometric information in computer vision has a long and rich history dating back to the seminal works of Lucas and Kanade [235] and Horn and Schunk [173]. The problem is usually formulated as a pairwise alignment of two images. One is the reference $\mathbf{I}_0$, while the other is the input $\mathbf{I}_1$. The two images are assumed to be related via a parametric transformation. The goal is to estimate the parameters of motion $\boldsymbol{p}$ such that the squared intensity error is minimized

$$\boldsymbol{p}^* = \operatorname*{argmin}_{\boldsymbol{p}} \sum_{\mathbf{u}\in\Omega_0} \frac{1}{2}\left\|\mathbf{I}_0(\mathbf{u}) - \mathbf{I}_1(\mathbf{w}(\mathbf{u};\boldsymbol{p}))\right\|^2, \tag{7.5}$$

where $\mathbf{u} \in \Omega_0$ denotes a subset of pixel coordinates in the reference image frame, and $\mathbf{w}\left(\cdot,\cdot\right)$ denotes the warping function [25]. Minimizing the photometric error has recently resurfaced as a robust solution to visual odometry (VO) from high frame rate imagery [107, 198, 345]. Notwithstanding, minimizing the photometric error has not yet been explored for the *joint* optimization of the motion and structure parameters for VSLAM in unstructured scenes.

The proposed approach fills in the gap by providing a photometric formulation for BA, which we call BA *without* correspondences.

7.3 Algorithm

BA is not limited to minimizing the reprojection error [372]. We reformulate the problem as follows. First, we assume an initial estimate of the camera poses $\boldsymbol{\theta}_i$ as required by geometric BA. However, we do not require tracking information for the 3D points. Instead, for every scene point $\boldsymbol{\xi}_j$, we assign a *reference* frame denoted by $r(j)$. The reference frame is used to extract a fixed square patch denoted by $\boldsymbol{\phi}_j \in \mathbb{R}^D$ over a neighborhood/window denoted by $\mathcal{N}$. In addition, we compute an initial *visibility* list indicating the frames where the point may be in view. The visibility list for the j^{th} point excludes the reference frame and is denoted by:

$$\mathbf{V}_j = \left\{k \;:\; k \neq r(j) \text{ and } \boldsymbol{\xi}_j \text{ is visible in frame } k\right\}, \text{ for } k \in [1, \dots, M]. \tag{7.6}$$

Given this information and the input images $\{\mathbf{I}_i\}_{i=1}^M$, we seek to estimate an optimal update to the motion $\Delta\boldsymbol{\theta}_i{}^*$ and structure parameters $\Delta\boldsymbol{\xi}_j{}^*$ that satisfy

$$\left\{\Delta\boldsymbol{\theta}_i^*, \Delta\boldsymbol{\xi}_j^*\right\} = \underset{\Delta\boldsymbol{\theta}_i, \Delta\boldsymbol{\xi}_j}{\operatorname{argmin}} \sum_{j=1}^{N} \sum_{k \in V(j)} \mathcal{E}(\boldsymbol{\phi}_j, \mathbf{I}_k; \Delta\boldsymbol{\theta}_k, \Delta\boldsymbol{\xi}_j), \tag{7.7}$$

where

$$\mathcal{E}(\boldsymbol{\phi}, \mathbf{I}'; \boldsymbol{\theta}, \boldsymbol{\xi}) = \sum_{\mathbf{u} \in \mathcal{N}} \frac{1}{2} \left\| \boldsymbol{\phi}(\mathbf{u}) - \mathbf{I}'(\pi(\boldsymbol{\theta}, \boldsymbol{\xi}) + \mathbf{u}) \right\|^2. \tag{7.8}$$

The notation $\mathbf{I}'(\pi(\cdot, \cdot) + \mathbf{u})$ indicates sampling the image intensities in a neighborhood about the current projection of the point. Since image projection results in subpixel coordinates, the image is sampled using an appropriate interpolation scheme (bilinear in this work). The algorithm is illustrated schematically in Fig. 7.1.

Linearization and sparsity

The optimization problem in Eq. (7.7) is nonlinear and its solution proceeds with standard techniques. Let $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ denote the current estimate of the camera and the scene point, and let the current projected pixel coordinate in the image plane be given by

$$\mathbf{u}' = \pi(\mathbf{T}(\boldsymbol{\theta}), \mathbf{X}(\boldsymbol{\xi})), \tag{7.9}$$

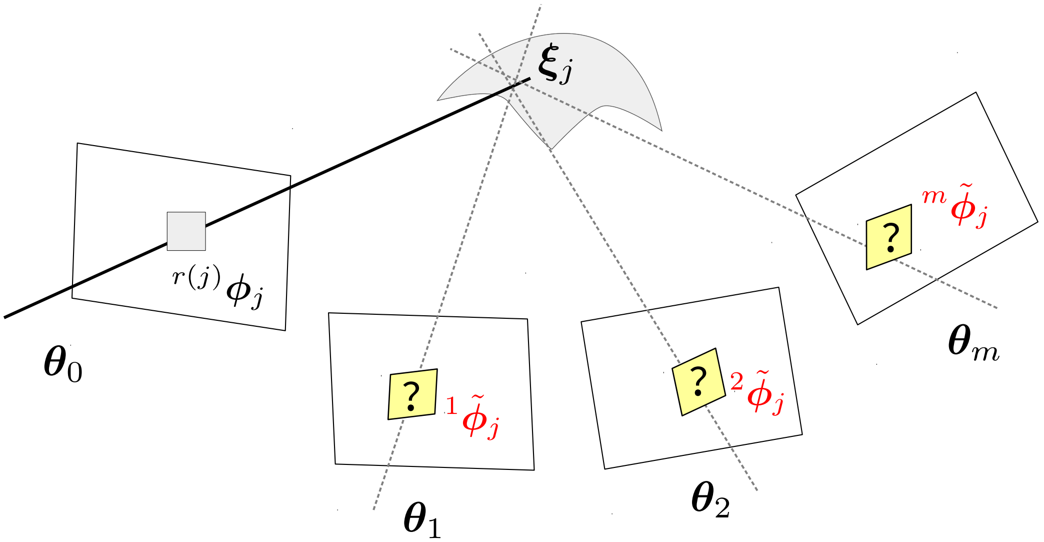

$\boldsymbol{\xi}_j$
$^{r(j)}\boldsymbol{\phi}_j$
$\boldsymbol{\theta}_0$
?
$^{1}\tilde{\boldsymbol{\phi}}_j$
$\boldsymbol{\theta}_1$
?
$^{2}\tilde{\boldsymbol{\phi}}_j$
$\boldsymbol{\theta}_2$
?
$^{m}\tilde{\boldsymbol{\phi}}_j$
$\boldsymbol{\theta}_m$

Figure 7.1: *Schematic of our approach. We seek to optimize the parameters of motion $\boldsymbol{\theta}_i$ and structure $\boldsymbol{\xi}_j$ such that the photometric error with respect to a fixed patch at the reference frame is minimized.*

then taking the partial derivatives of the 1st-order expansion of the photometric error in Eq. (7.8) with respect to the parameters we obtain

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{\theta}} = \sum_{\mathbf{u}\in\mathcal{N}} \mathbf{J}^\top(\boldsymbol{\theta}) \left|\phi(\mathbf{u}) - \mathbf{I}'(\mathbf{u}' + \mathbf{u}) - \mathbf{J}(\boldsymbol{\theta})\Delta\boldsymbol{\theta}\right| \tag{7.10}$$

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{\xi}} = \sum_{\mathbf{u}\in\mathcal{N}} \mathbf{J}^\top(\boldsymbol{\xi}) \left|\phi(\mathbf{u}) - \mathbf{I}'(\mathbf{u}' + \mathbf{u}) - \mathbf{J}(\boldsymbol{\xi})\Delta\boldsymbol{\xi}\right|, \tag{7.11}$$

where $\mathbf{J}(\boldsymbol{\theta}) = \nabla\mathbf{I}(\mathbf{u}' + \mathbf{u})\frac{\partial \mathbf{u}'}{\partial \boldsymbol{\theta}}$, and $\mathbf{J}(\boldsymbol{\xi}) = \nabla\mathbf{I}(\mathbf{u}' + \mathbf{u})\frac{\partial \mathbf{u}'}{\partial \boldsymbol{\xi}}$. The partial derivatives of the projected pixel location with respect to the parameters are identical to those obtained when minimizing the reprojection error in Eq. (7.1), and $\nabla\mathbf{I} \in \mathbb{R}^{1\times 2}$ denotes the image gradient. By equating the partial derivatives in Eqs. (7.10) and (7.11) to zero we arrive at the normal equations which can be solved using standard methods [289].

We note that the Jacobian involved in solving the photometric error has a higher dimensionality than its counterpart in geometric BA. This is because the dimensionality of intensity patches ($D \geq 3 \times 3$) is usually higher than the dimensionality of feature projections (typically 2 for a monocular reconstruction problem). Nonetheless, the Hessian remains *identical* to minimizing the reprojection error and the linear system remains sparse and is efficient to decompose. The sparsity pattern of the photometric BA problem is illustrated in Fig. 7.2.

Another important note is that since the parameters of motion and structure are refined jointly, the location of the patch at the reference frame $\phi(\mathbf{u})$ in Eq. (7.8) will additionally depend on the pose parameters of the reference frame. Allowing the reference patch to "move" during the optimization adds additional terms to the Hessian (additional terms will appear in the motion parameters blocks of the Hessian, these are shown at left hand corner of the Hessian in Fig. 7.2). In terms of computational complexity, the additional runtime from allowing the reference patch to move is minimal as the algorithm is implemented in a sliding window fashion. However, including inter–pose dependencies is undesirable as, depending on the initialization quality, the location of the reference patch might drift. For instance, we might introduce a biased solution where the patches drift to image regions with brighter absolute intensity values in an attempt to obtain the minimum energy in low-texture areas.

To address this problem, we fix the patch appearance at the reference frame by storing the patch values as soon as the reference frame is selected. This is equivalent to assuming a known patch appearance from an independent source. Under this assumption, the optimization problem now becomes: given a known and fixed patch appearance of a 3D point in the world, refine the parameters of the structure and motion such that photometric error between the fixed patch and its projection onto the other frames is minimized. This assumption has two advantages: (1) the Hessian sparsity pattern remains identical to the familiar form when minimizing the reprojection error using traditional BA, and (2) we can refine the three coordinates (or the full four projective coordinates [372]) of the scene points as opposed to only refining depth along a fixed ray in space.

In addition to improving the accuracy of VSLAM, the algorithm does not require extensive parameter tuning. This is now possible by allowing the algorithm to determine the correct correspondences, hence eliminating the many steps required to ensure outlier-free correspondences with traditional BA. The current implementation of the proposed algorithms is controlled by the three parameters summarized in Table 7.1 and explained next.

The current implementation of our algorithms is controlled by three parameters summarized in Table 7.1 and explained next.

Selecting pixels

While it is possible to select pixel locations at every frame using a standard feature detector, such as Harris [158] or FAST [311], we opt to use a simpler and more efficient strategy based on the gradient magnitude of the image. This is performed by selecting pixels with local maxima in a 3×3 neighborhood of the absolute gradient magnitude of the image.

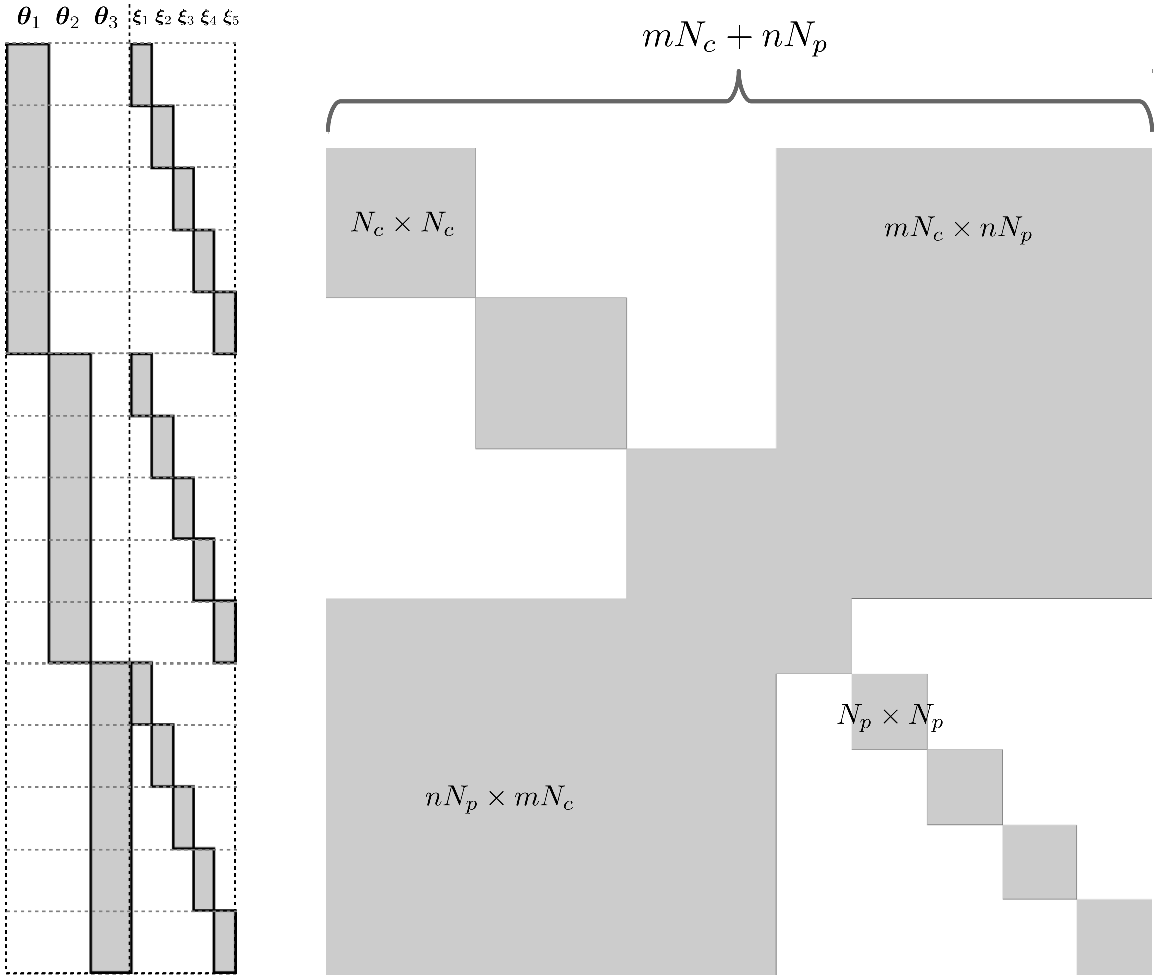

$\boldsymbol{\theta}_1$ $\boldsymbol{\theta}_2$ $\boldsymbol{\theta}_3$ ξ_1 ξ_2 ξ_3 ξ_4 ξ_5
$mN_c + nN_p$
$N_c \times N_c$
$mN_c \times nN_p$
$N_p \times N_p$
$nN_p \times mN_c$

Figure 7.2: *Shown on the left is the form of the Jacobian for a photometric bundle adjustment problem consisting of* 3 *cameras,* 4 *points, and using a* 9*-dimensional descriptor, with $N_c = 6$ parameters per camera, and $N_p = 3$ parameters per point. Form of the normal equations is shown on the right. The illustration is not to scale across the two figures.*

Table 7.1: *Configuration parameters for our proposed algorithm shown in Algorithm 3.*

Parameter	Value
Patch radius	1 or 2
Non max supp. radius	1
Max distance to update V_j	2

The rationale is that pixels with vanishing intensity gradients do not contribute to the linear system in Eqs. (7.10) and (7.11). Hence, pixels with larger gradients are preferable as they indicate a measure of textureness [329]. Other strategies for pixel selection could be used [98, 256]; however, we found that the current scheme works well as it ensures an even distribution of coordinates across the field of view of the camera [288].

In image-based (photometric) optimization there is always a distinguished reference frame providing fixed measurements [182, 285, 344]. Selecting a single reference in photometric VSLAM is unnecessary, and may be inadvisable. It is unnecessary as the density of reconstruction is not our main goal. It is inadvisable because we need the scene points to serve as tie points [7] and form a strong network [372]. Given the nature of camera motion in VSLAM selecting points from every frame ensures the strong network of connections between the tie points. For instance, typical hand-held, and ground robots motions are mostly forward without points leaving the field of view quickly.

Selecting new scene points at every frame using the aforementioned non maxima suppression procedure has one caveat. If we always select pixels with strong gradients between consecutive frames, then we are likely to track previous scene points rather than finding new ones. This is because pixels with locally maximum gradient magnitude at the consecutive frame are most likely images of previously selected points. Treating projections of previously initialized scene points as new observations is problematic because it introduces unwanted dependencies in the normal equations and superficially increases the number of independent measurements in the linearized system of equations.

To address this issue, we assume that the scene and motion initialization is accurate enough to predict the location of current scenes in the new frame. Prior to initializing new scene points, we use the provided pose initialization to warp all previously detected scene points that are active in the optimization sliding window onto the new frame. After that, we mark a 3×3 square area at the projection location of the previous scene points as an invalid location for selecting new points. This step is illustrated in Fig. 7.3, and is best summarized in our pseudo code shown in Algorithm 3.

The number of selected points per frame varies depending on the image resolution and texture information in the image. In our experiments, this number ranges between ≈ 4000–10000 points per image.

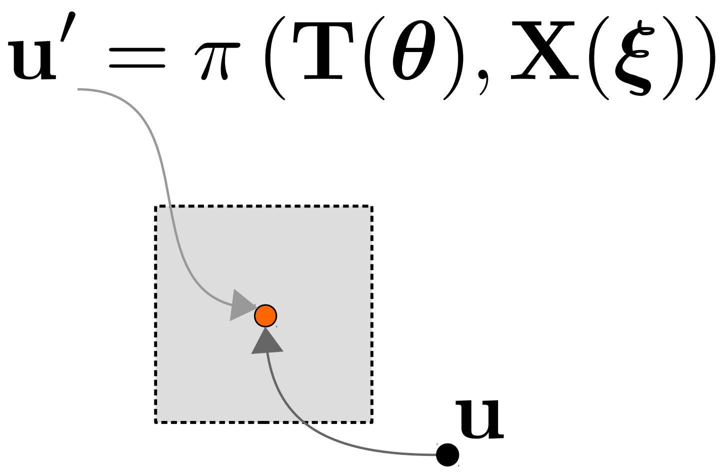

$\mathbf{u}' = \pi\left(\mathbf{T}(\boldsymbol{\theta}), \mathbf{X}(\boldsymbol{\xi})\right)$
$\mathbf{u}$

Figure 7.3: *Illustration of how we avoid reinitializing the same point at a new frame. Using the pose initialization of the new frame $\boldsymbol{\theta}$, we project previous scene points and reserve a 3×3 area where new scene points cannot be initialized.*

Determining visibility

Ideally, we would like to assume that newly initialized scene points are visible in all frames and to rely on the algorithm to reliably determine if this is the case. However, automatically determining the visibility information along with structure and motion parameters is challenging, as many scene points quickly go out of view, or become occluded. Their inclusion in the optimization problem incurs an unnecessary computational complexity, reduces robustness, and increases the uncertainty of the solution.

An efficient and reliable measure to detect occlusions and points that cannot be tracked reliably is the normalized correlation. For all scene points that are close to the current frame i, we use the pose initialization $\mathbf{T}_i$ to extract a 5×5 intensity patch. The patch is obtained by projecting the scene points to the new frame and its visibility list is updated if the zero-mean normalized correlation score (ZNCC) is greater than 0.8. We allow ± 2 frames for a point to be considered close, *i.e.* $|i - r(j)| \leq 2$. This procedure is similar to determining the visibility information in multi-view stereo algorithms [132] and is best summarized in Algorithm 3.

Optimization details

We use the Ceres optimization library [6], designed for BA problems, to optimize the objective in Eq. (7.7). We use the Levenberg-Marquardt algorithm [222, 245] to minimize a

Algorithm 3 Summary of image processing in bundle adjustment without correspondences.

```
procedure ProcessFrame($\mathbf{I}_i, \mathbf{T}_i$)
    Step 1: establish connections to the new frame
    mask = all_valid(rows($\mathbf{I}$), cols($\mathbf{I}$))
    for all scene points $\mathbf{X}_j$ in sliding window do
        if reference frame $r(j)$ is too far from $i$ then
            continue
        $\mathbf{x} \coloneqq$ projection of $\mathbf{X}_j$ onto image $\mathbf{I}_i$ using pose $\mathbf{T}_i$
        $\boldsymbol{\phi}' \coloneqq$ patch at $\mathbf{x}$ and $\boldsymbol{\phi} \coloneqq$ reference patch for $\mathbf{X}_j$
        if zncc($\boldsymbol{\phi}, \boldsymbol{\phi}'$) $>$ threshold then
            add frame $i$ to visibility list $V_j$
            mask($\mathbf{u}$) = invalid

    Step 2: add new scene points
    $\mathbf{G} \coloneqq$ gradient magnitude of $\mathbf{I}_i$
    for all pixels $\mathbf{u}$ in $\mathbf{I}_i$ do
        if $\mathbf{u}$ is a local maxima in $\mathbf{G}$ then
            if location $\mathbf{u}$ is valid in mask then
                initialize a new point $\mathbf{X}$ with reference patch at $\mathbf{I}(\mathbf{u})$
```

Huber loss function instead of squared loss to improve robustness. Termination tolerances are set to 1×10^{-6}, and automatic differentiation facilities are used. Image gradients used in the linearized system in Eqs. (7.10) and (7.11) are computed using a central difference filter given by $\frac{1}{2}\,[-1,\ 0, 1]$. Finally, we make explicit use of the Schur complement to obtain a more efficient solution [402].

Since scene points do not remain in view for an extended period in most VSLAM datasets, the photometric refinement step is performed using a sliding window of five frames [355]. The motion parameters of the first frame in the sliding window is held constant to fixate the Gauge freedom [372]. The 3D parameters of the scene points in the first frame, however, are included in the optimization.

7.4 Experiments

In this section, we evaluate the performance of the proposed algorithm on two commonly used VSLAM benchmarks to facilitate comparisons with the state-of-the-art. The first is the KITTI benchmark [138], which contains imagery from an outdoor stereo camera mounted on a vehicle. The second is the Malaga dataset [46], which is particularly challenging for

VSLAM because the baseline of the camera (12cm) is small relative to the scene structure.

7.4.1 The KITTI Benchmark

Initializing with geometric BA

Torr and Zisserman [369] convincingly argue that the estimation of structure and motion should proceed by feature extraction and matching to provide a good initialization for BA-based refinement techniques. Here, we use the output of ORB-SLAM [276], a recently proposed state-of-the-art VSLAM algorithm, to initialize our method. ORB-SLAM not only performs geometric BA, but also implements loop closure to improve accuracy. The algorithm is currently one of the top performing algorithms on the KITTI benchmark [138].

We only use the pose initialization from ORB-SLAM. We do not make use of the refined 3D points as they are available at selected keyframes only. This is because images in the KITTI benchmark are collected at 10 Hz, while the vehicle speed exceeds 80 km/h in some sections. Subsequently, the views are separated by a large baseline, which violates the small displacement assumption required for the validity of linearization in Eqs. (7.10) and (7.11).

Hence, to initialize 3D points we use the standard block matching stereo algorithm implemented in OpenCV. This is a winner-takes-all brute force search strategy based on the sum of absolute intensity differences (SAD). The algorithm is configured to search for 128 disparities using a 7×7 aggregation window and a left–right consistency check.

The choice of initializing the algorithm with ORB-SLAM is intentional to assess the accuracy of the algorithm in comparison to the Gold Standard solution from traditional BA. We note, however, that we could use LSD-SLAM [110] to obtain a VSLAM system without correspondences at all. In fact, initial pose estimates could be provided by external sensors, such as low quality GPS.

Performance of the algorithm is shown in Fig. 7.4 and not only does it outperform the accuracy of (bundle adjusted and loop closed) ORB-SLAM, but also it outperforms other top performing algorithms, especially in the accuracy of estimating rotations. Compared algorithms include: ORB-SLAM [276], LSD-SLAM [107, 110], VoBA [361], and MFI [23].

We note that sources of error in our algorithm are correlated with faster vehicle speeds. This is to be expected as the linearization of the photometric error holds only in a small neighborhood. This could be mitigated by implementing the algorithm in scale-space [224], or improving the initialization quality of the scene structure (either by better stereo, or bet-

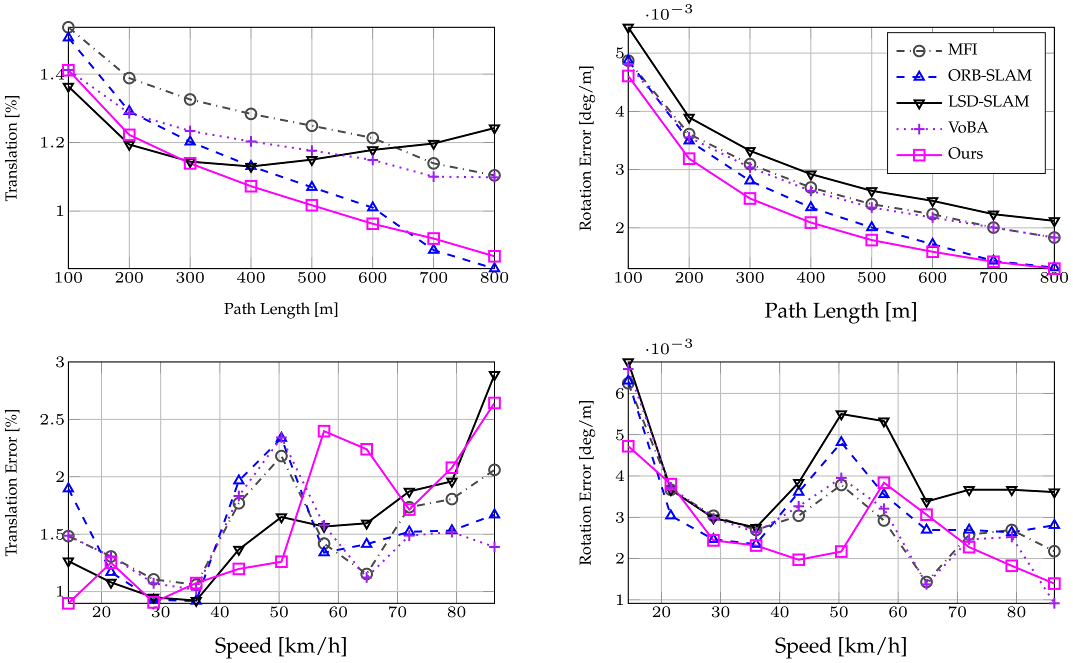


Figure 7.4: *Comparison to state-of-the-art algorithms on the KITTI benchmark. Our approach performs the best. Error in our approach correspond to segments of the data when the vehicle is driving at a high speed, which increases the magnitude of motion between frames and affects the linearization assumptions. No loop closure, or keyframing is performed using our algorithm. Improvement is shown qualitatively in Fig. 7.5.*

ter scene points obtained from a geometric BA refinement step). Interestingly, however, the rotation error is reduced at high speeds which can be explained by lack of large rotations. The same behavior can be observed with LSD-SLAM's performance as both methods rely on the photometric error, but our rate of error reduction is higher due to the joint refinement of pose and structure parameters.

Initializing with frame–frame VO

Surprisingly, and contrary to other image-based optimization schemes [95, 133], our algorithm does not require an accurate initialization to be useful. Fig. 7.6 demonstrates a significant improvement in accuracy when the algorithm is initialized using frame–frame VO estimates with unbounded drift [15].

Interestingly, however, when starting from a poor initialization our algorithm does not attain the same accuracy as when initialized using a better quality starting point as shown in Fig. 7.4. This leads us to conclude the algorithm may be sensitive to the initialization conditions. Importantly, however, the algorithm is able to improve upon a poor initializa-

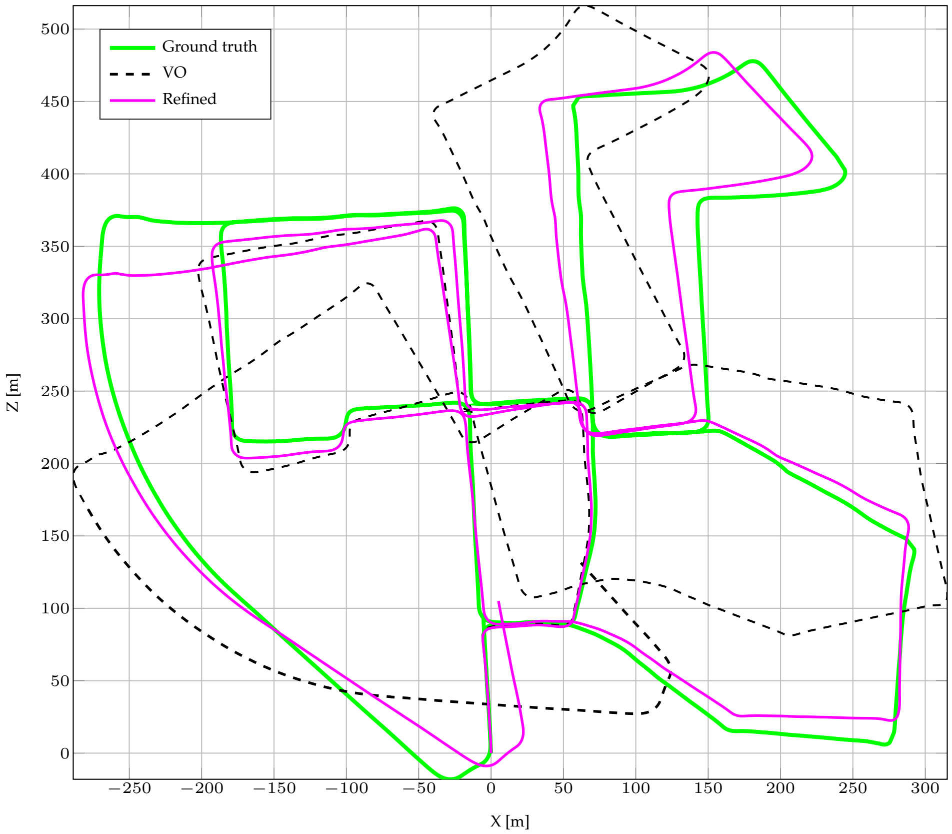


Figure 7.5: *Magnitude of improvement starting from a poor initialization shown on the first sequence of the KITTI benchmark. Quantitative evaluation is shown in Fig. 7.4. We used a direct (correspondence-free) frame–frame VO method to initialize the pose parameters [15] (Chapter 6).*

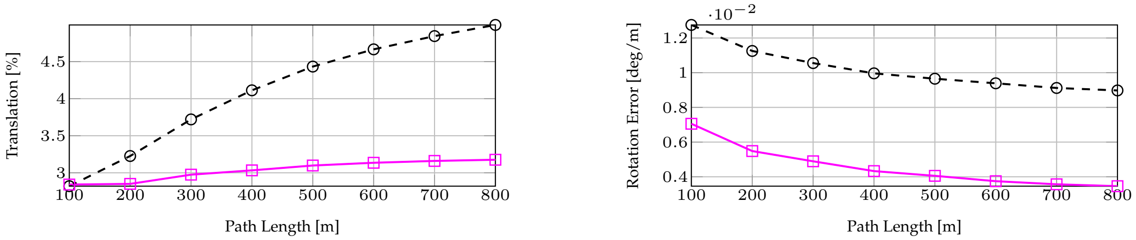


Figure 7.6: *Quantitative easement of accuracy improvement in using poor initialization.*

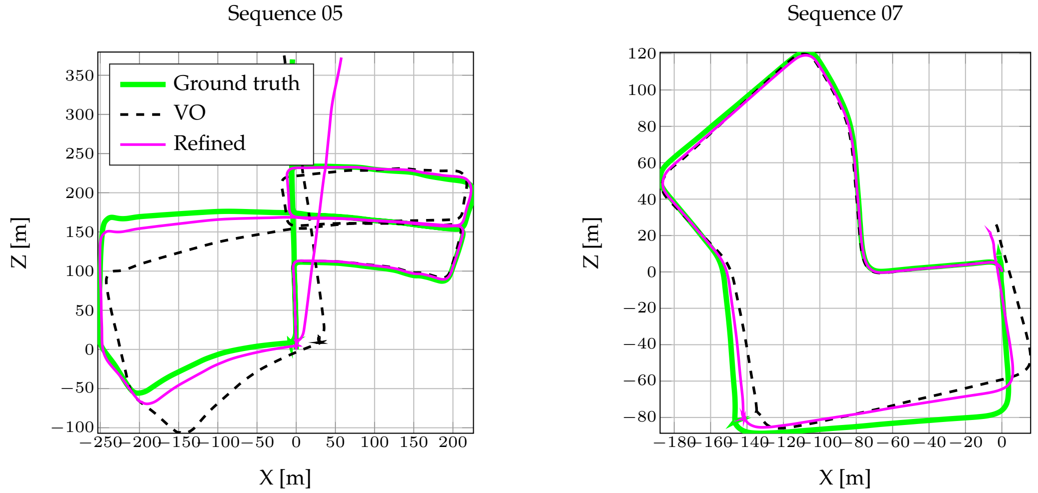


Figure 7.7: *Improvement on poor initialization shown for KITTI sequences* 5 *and* 7*.*

tion as shown in Figs. 7.7 to 7.10.

Convergence characteristics and runtime

As shown in Fig. 7.11 most of the photometric error is eliminated in the first five iterations of the minimization problem. While this is by no means a metric of quality, it is reassuring as it indicates a well-behaved minimization procedure.

After the first five iterations, the rate of the relative reduction in error slows down. This may be related to using linear interpolation to evaluate the photometric error, or the use of central differences to estimates gradients. Higher order interpolation methods [378], or more accurate image gradients [117] could have an influence on the rate of convergence and remain to be explored.

The number of iterations and cumulative runtime per sliding window of 5 frames is shown in Fig. 7.12. The median number of iterations is 34 with a standard deviation of ≈ 6. Statis-

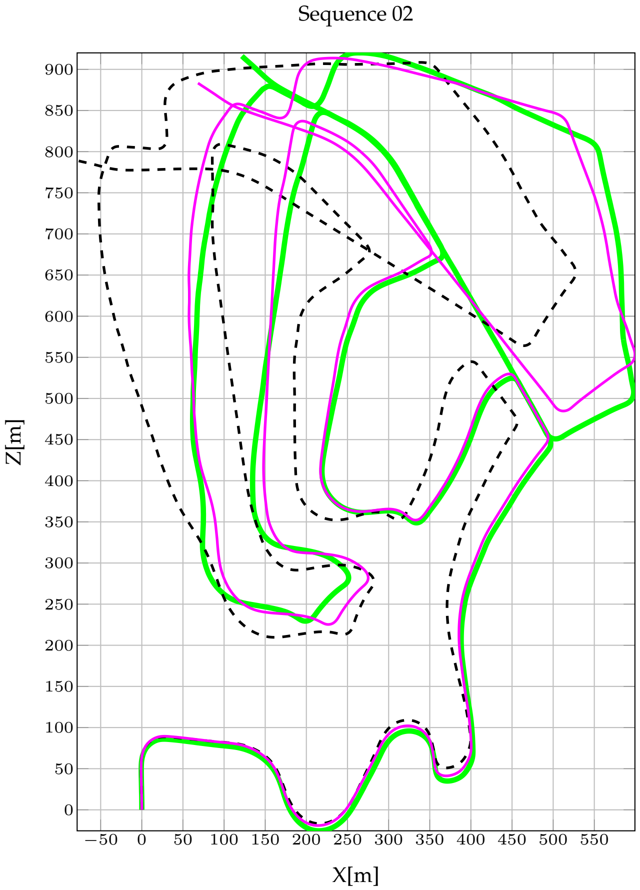


Figure 7.8: Improvement on poor initialization shown for KITTI sequence 2.

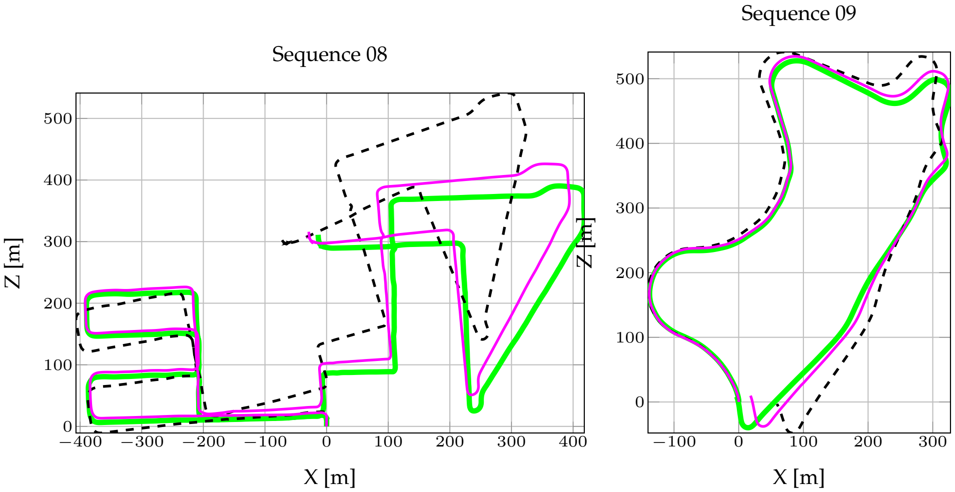


Figure 7.9: Improvement on poor initialization shown for KITTI sequences 8 and 9.

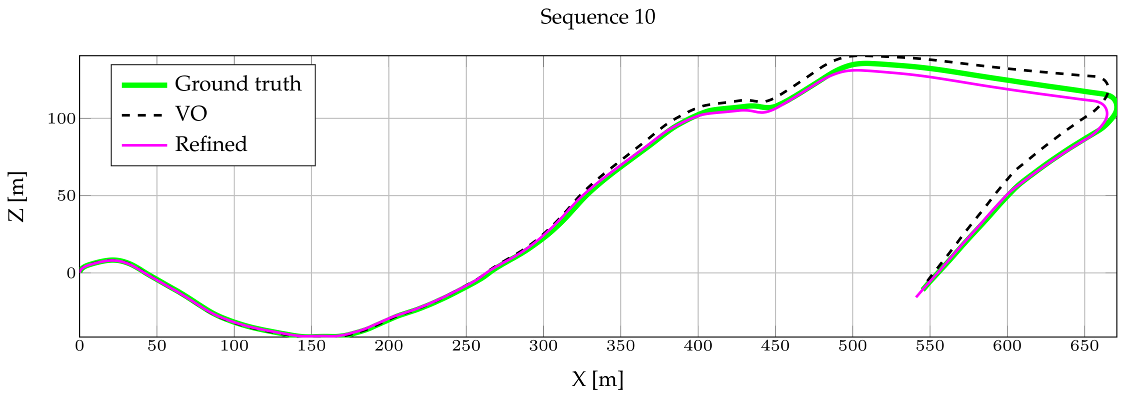


Figure 7.10: *Improvement on poor initialization shown for KITTI sequence* 10.

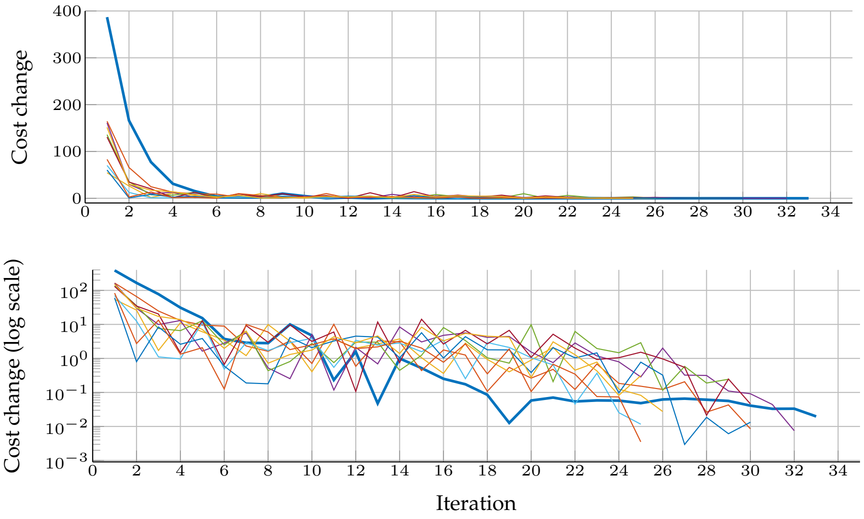


Figure 7.11: *Rate of error reduction at every iteration shown for the first* 10 *sliding windows, each with* 5 *frames. The thicker line shows the first bundle, which has the highest error. Most of the error is eliminated with the first five iterations. The bottom figure is shown in log scale for more details.*

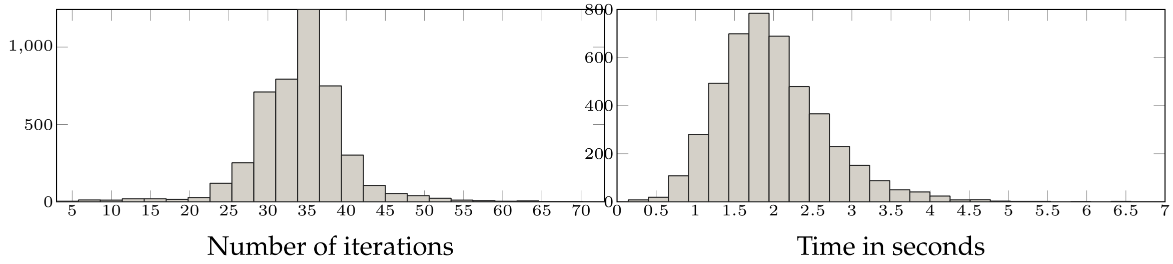


Figure 7.12: *Histogram of the number of iterations (on the left) and runtime (on the right). The median number of iterations is* 34*, with a standard deviation of* 6.02*. The median run time is* 1.89*, mean* 1.98 *and standard deviation of* 0.69*. The runtime is reported for sliding window of five frames.*

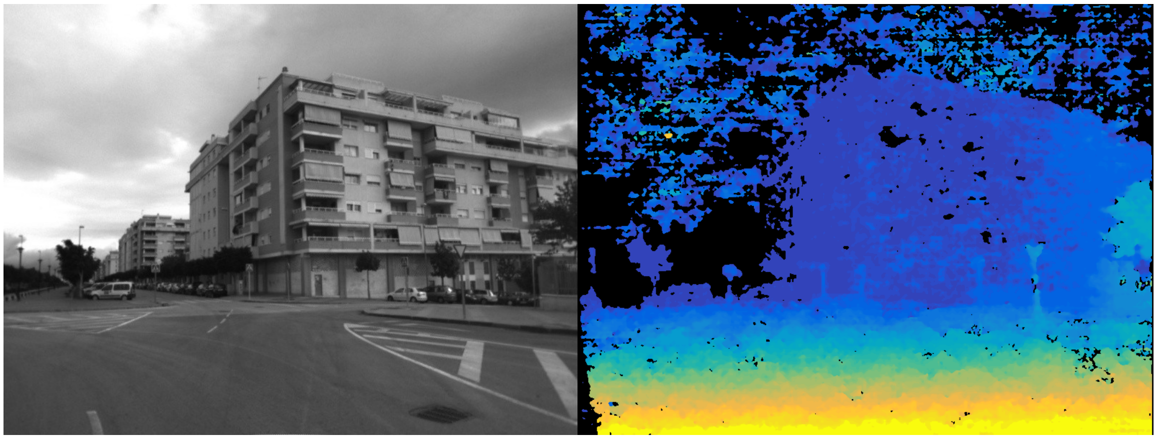


Figure 7.13: Quality of stereo used to initialize our algorithm on the Málaga dataset. The pixels marked in black indicate missing disparity estimates.

tics are computed on the KITTI dataset frames. The runtime is $\approx 2s$ per sliding window ($400\,\mathrm{ms}$ per frame) using a laptop with a dual core processor clocked at 2.8 GHz and 8 GB of RAM, which limits parallelism. We note that it is possible to improve the runtime of the proposed method significantly using the CPU, or the GPU. The bottleneck of the proposed algorithm is image interpolation (which can be done efficiently with SIMD instructions) and the reliance on automatic differentiation (which limits any code optimization as the code must remain simple for automatic differentiation to work).

7.4.2 The Málaga Stereo Dataset

The Málaga dataset [23] is a particularly challenging dataset for VSLAM. The dataset features driving on city roads using a small baseline stereo camera at resolution 800×600. The baseline of the stereo is 12 cm which provides little parallax for resolving distal scene points. In addition, the camera is pointed upward toward the sky to avoid imaging the vehicle, which limits using points on the ground plane and closer to the camera. We use extracts, 1, 3, and 6 in our evaluation.

Our experimental setup is similar to the KITTI dataset. However, we estimate the stereo using the SGM algorithm [168], as implemented in OpenCV. The stereo is used to estimate 16 disparities with a SAD block size of 5×5. The quality of stereo is low due to the difficulty of the dataset as shown in Fig. 7.13. We did not observe a significant difference in performance when using block matching instead of SGM.

The Malaga dataset provides GPS measurements, but they are not accurate enough for

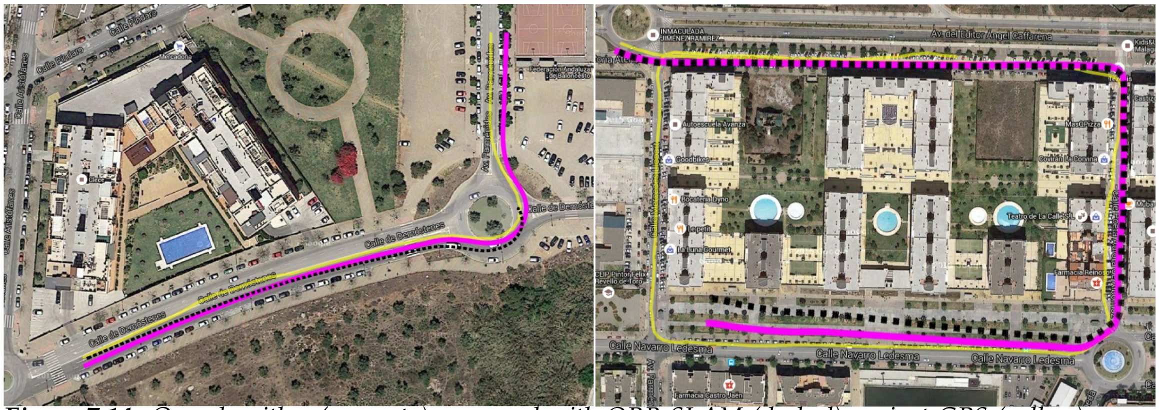


Figure 7.14: *Our algorithm (magenta) compared with ORB-SLAM (dashed) against GPS (yellow) on extracts* 3 *and* 6 *of the Malaga dataset. For extract* 3 *ORB-SLAM loses the tracking during the roundabout, where our algorithm continues without an initialization. Results for extract* 6 *are shown up to frame* 3000 *as ORB-SLAM looses tracking then. The figure best viewed in color. (Maps courtesy of Google Maps.)*

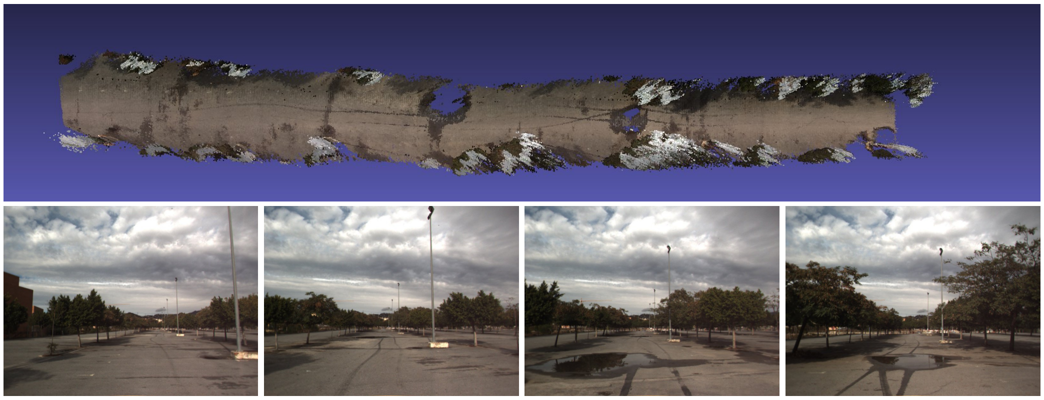


Figure 7.15: *Dense map from Malaga dataset extract* 1.

quantitative evaluation. The GPS path, however, is sufficient to qualitatively demonstrate precision. Results are shown in Fig. 7.14 in comparison with ORB-SLAM [276], which we used its pose estimates to initialize our algorithm. We note that in extract 3 of the Malaga dataset (shown on the left in Fig. 7.14), ORB-SLAM loses tracking during the turn and our algorithm continues *without* initialization.

To assess the quality of pose estimates we demonstrate results on a dense reconstruction procedure shown in Figs. 7.15 and 7.16. Using the estimated camera trajectory, we chain the first 6 m of the disparity estimates to generate a dense map. As shown in Fig. 7.15, the quality of pose estimates appears to be good.

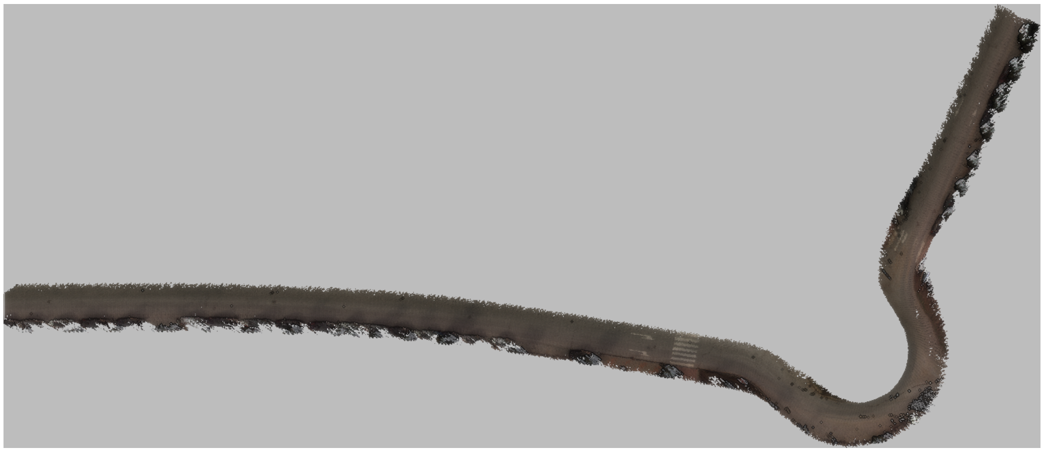


Figure 7.16: *Dense map from Malaga dataset extract* 3.

7.5 Related Work

Geometric BA

BA has a long and rich history in computer vision, photogrammetry and robotics [372]. BA is a large geometric minimization problem with the important property that variable interactions result in a *sparse* system of linear equations. This sparsity is key to enabling large scale applications [4, 210]. Exploiting this sparsity is also key to obtaining precise results efficiently [111, 186]. The efficiency of BA has been an important research topic especially when handling large datasets [287, 398] and in robotics applications [189, 190, 209]. Optimality and convergence properties of BA have been studied at length [160, 161, 191] and remain of interest to date [2]. All the aforementioned research could be integrated into our framework.

Direct multi-frame alignment

By direct alignment we mean algorithms that estimate the parameters of interest from the image data directly and without relying on sparse features as an intermediate representation of the image [181]. The fundamental differences between direct methods (like the one proposed herein) and the commonly used feature-based pipeline is how the correspondence problem is tackled and is not related to the density of the reconstruction.

In the feature-based pipeline [369], structure and motion parameters are estimated from known, pre-computed and fixed correspondence. In contrast, the direct pipeline to motion estimation does not used fixed correspondences. Instead, the correspondences are estimated as a byproduct of directly estimating the parameters of interest (*e.g*. structure and motion).

The use of direct algorithms for SFM applications was studied for small–scale problems [171, 243, 291, 344], but feature-based alignment has proven more successful in handling wide baseline matching problems [369] as small pixel displacements is an integral assumption for direct methods. Nonetheless, with the increasing availability of high frame-rate cameras, video applications, and increasing computational power, direct methods are demonstrating great promise [110, 198, 285]. For instance, direct estimation of motion from RGB-D data was shown to be robust, precise and efficient [198, 206, 345].

To date, however, the use of direct methods in VSLAM has been limited to frame–frame motion estimation (commonly referred to as visual odometry). Approaches that make use of multiple frames are designed for dense depth estimation only and multi-view stereo [132, 285], which assume a correct camera pose and only refine the scene structure. Other algorithms can include measurements from multiple frames, but rely on the presence of structures with strong planarity in the environment [182, 335] (or equivalently assuming a rotation only motion such that the motion of the camera can be represented as a homography [233]).

In this work, in contrast to previous research in direct image-based alignment [285, 344], we show that provided good initialization, it is possible to jointly refine the structure and motion parameters by minimizing the photometric error and without restricting the camera motion or the scene structure.[1]

The LSD-SLAM algorithm [110] is a well-known recently proposed direct algorithm for vision-based motion estimation. In comparison to our work, the fundamental difference is that we refine the parameters of motion and structure jointly in one large optimization problem. In LSD-SLAM, the photometric error is used to estimate the motion, while scene (inverse) depth is estimated using small baseline stereo with fixed camera fixed. The joint optimization of motion and structure proposed herein is important in future work concerning the optimality and convergence properties of photometric structure-from-motion (SFM) and photometric, or direct, VSLAM. Our work can be regarded as an extension of LSD-SLAM where the parameters of motion and structure are refined jointly.

[1] While a conference publication based on this chapter was under peer review [13], Engel et al. [108] proposed a photometric VSLAM approach similar to the work presented in this chapter and our prior thesis proposal document [9].

Dense multi-view stereo (MVS)

MVS algorithms aim at recovering a dense depth estimate of objects or scenes using many images with known pose [132]. To date, however, research on simultaneous refinement of motion and depth from multiple frames remains sparse. Furukawa and Ponce [133] were among the first to demonstrate that relying on minimizing the reprojection error is not always accurate enough. The work demonstrates that calibration errors could have a large impact on accuracy. Furukawa and Ponce address this problem by refining the correspondences using photometric information and visibility information from an intermediate dense reconstruction in a guided matching step. The process is then interleaved with traditional geometric BA using the improved correspondences to obtain a better reconstruction accuracy. In our work, however, we show that interleaving the minimization of the photometric error with the reprojection error may be unnecessary and solving the problem directly is feasible.

Recently, Delaunoy and Pollefeys [95] proposed a photometric BA approach for dense MVS. Starting from a precise initial reconstruction and a mesh model of the object, the algorithm is demonstrated to enhance MVS accuracy. The imaging conditions, however, are ideal and brightness constancy is assumed [95]. In our work, we do not require a very precise initialization and can address challenging illumination conditions. More importantly, the formulation proposed by Delaunoy and Pollefeys requires the availability of an accurate dense mesh, which is not possible to obtain in VSLAM scenarios. Furthermore, initialization requirements appear to be much higher than our approach.

7.6 Summary

In this work, we show how to improve on the accuracy of the state-of-art VSLAM methods by minimizing the photometric error across multiple views. In particular, we show that it is possible to improve results obtained by minimizing the reprojection error in a bundle adjustment (BA) framework. We also show, contrary to previous image-based minimization work [107, 110, 285, 344, 345], that the *joint* refinement of motion and structure is possible in unconstrained scenes without the need for alternation or disjoint optimization.

The accuracy of minimizing the reprojection using traditional BA is limited by the precision and accuracy of feature localization and matching. In contrast, our approach — BA without correspondences — determines the correspondences implicitly such that the photometric consistency is maximized as a function of the scene structure and camera motion parameters.

Finally, we show that accurate solutions to geometric problems in vision are not restricted to geometric primitives such as corners and edges, or even planes. We look forward to more sophisticated modeling of the geometry and photometry of the scene beyond the intensity patches used in our work.

CHAPTER 8

Summary and Conclusions

Current progress in geometric estimation problems is a testament to the immense research advances in Computer Vision. Its effect on robotic applications can be readily seen in our daily lives and is rapidly approaching the ranks of a mature technology in many application domains. In this work, we contributed algorithms and representations aimed at improving the robustness of geometric estimation problems in face of adverse imaging conditions.

In Vision, there are two main paradigms for estimating geometric quantities from images. The first is what is commonly known as the feature-based approach [369] exemplified by minimizing the reprojection error and is commonly employed in structure-from-motion (SFM) applications. The second is what is known as direct methods [181] exemplified by the Lucas and Kanade algorithm [235] for iterative image alignment. The distinction between the two paradigms is how the *correspondences problem* is tackled.

Feature-based approaches to motion estimation reduce the problem to a purely geometric perspective. Given a set of fixed and known feature correspondences in the image plane, pose estimation is formulated using the geometric constraints. The approach is a natural embodiment of the geometry underlying image formation and optics. Once these correspondences are known, well-established tools form projective geometry in conjunction with robust estimation frameworks, such as RANSAC [123], are readily available to address a range of pose estimation problems [119, 160].

The most challenging part of the feature-based pipeline is the abstraction of the image into a few keypoint positions that must be matched with high precision. First, interest points must be detected and localized with high sub-pixel accuracy. Second, the interest points must be matched despite changes in viewpoint and appearance. The difficulty of this two-step approach, especially the detection step, increase measurably with degradations of the image quality. The matching step, however, is easier to accomplish with the plethora of distinctive local feature descriptors.

On the other hand, direct methods as exemplified by the Lucas and Kanade algorithm for parametric image alignment [235], and the Horn and Schunk algorithm for the estimation of the optical flow [173] were among the first techniques for geometric estimation problems in Computer Vision. In the direct approach correspondences are unknown and must be established as a byproduct of pose estimation. As direct method can make use of high- and low-frequency image content alike, they are generally regarded as more precise than their sparse feature-based counterparts [181]. Nonetheless, direct methods are highly susceptible to changes in illumination and thus lose their desirable properties in face of commonly accruing photometric variations. Addressing the range of photometric variations commonly encountered in robotic applications using consumer cameras is a challenging task when assumptions about the scene and the illumination source cannot be established.

In this work, we combine the best of both feature-based and direct algorithms to improve the robustness of vision-only pose estimation in face of adverse imaging conditions characterized by poor illumination. We made use of illumination-invariant binary feature descriptors in a direct alignment framework to overcome sudden and drastic changes in illumination. In contrast to current techniques, the proposed method makes little to no assumptions about the structure of the world or the type of illumination, hence providing a nonparametric means to handling challenging appearance variations. In addition, due to the compactness of binary descriptors, the approach is shown to work faster than real-time when high frame-rate video data are available.

The direct approach using the binary descriptor constancy proposed in this work was shown to work robustly and efficiently in light of sudden and drastic changes in illumination as demonstrated on region-based tracking problems, such as template tracking and more challenging forms of pose estimation such as visual odometry. In other pose estimation problems, such as vision-based simultaneous localization and mapping (VSLAM), where not only camera pose estimates are required but also the scene structure, direct methods fall behind in comparison to the feature-based pipeline. This is because, to date, there has been no demonstration of direct VSLAM that is able to make use of multiple-views to reduce the accumulated drift without making assumptions about the structure of the scene, such as planarity, or restricted to the motion of the camera to, for instance, rotation only.

In this work, we developed a correspondence-free bundle adjustment algorithm based on the maximization of the photometric consistency across multiple-views. In contrast to prior work, our formulation refines the motion and structure parameters *jointly* without the need for alternation and without imposing special requirements on the scene or the camera motion. We evaluated this novel approach on a range of outdoor stereo datasets and showed that a photometric refinement technique can improve upon results obtained from the state-of-the-art algorithms, even those obtained using loop closure constraints.

All in all, we demonstrated two main points in this dissertation. First, we established the utility of dense image alignment techniques in combination with feature descriptors for robust and efficient pose estimation in challenging environments. Second, and more interestingly, while state-of-the-art VSLAM have matured, there are still sources of errors left unmodeled by minimizing the reprojection error and loop closure. We hypothesized that some sources of error could be attributed to inaccuracies in feature localization as most commonly used feature detectors are optimized for detection and not accurate sub-pixel localization required for accurate geometric estimation. To address these limitations we developed a correspondence-free photometric solution capable of correcting some of these errors by bringing back the image in the loop.

Methods for direct estimation of pose (except for optical flow estimation) have been relatively understudied in comparison to the feature-based pipeline. In the next chapter, we outline possible research directions for the future with focus on direct motion estimation for robotic applications, such as visual odometry and VSLAM. We further discuss some theoretical considerations pertaining to direct motion estimation which heretofore have not been answered.

CHAPTER 9

Future Work

> The only reason for time is so that everything does not happen at once.
>
> Albert Einstein

While vision-only pose estimation has advanced immensely there are several avenues of future work with emphasis on photometric, or direct, techniques which have been relatively understudied in comparison to feature-based methods. Given the continued availability of massively parallel hardware and the continuous development of specialized hardware targeting computer vision applications, direct methods for pose estimation will become more mainstream as they are trivially parallelizable and are potentially more precise as all image areas with non-vanishing gradients could in principle be used for pose estimation. In addition, as the community is gravitating towards deep learning and end–end pose and correspondences estimation [122, 370, 392, 401], gradient-based alignment of feature descriptors will become more popular [127, 226, 227, 394, 401]. Some avenues of future research include the following:

- The use of direct methods for motion estimation has been on the rise, especially for visual odometery. It has been demonstrated on two interesting variations to date. The first is visual-inertial navigation [47, 294, 380]. The other is motion estimation under rolling-shutter artifacts [200, 203, 258]. However, a fundamental problem that has not been studied in a direct alignment framework is the effect and correction of mis-calibration errors, especially nonlinear lens distortion. In this dissertation, we assumed that the camera has been intrinsically calibrated with sufficient accuracy. However, leaving calibration errors unmodeled has a higher impact on the robustness and the accuracy of direct methods in comparison to the feature-based pipeline. If the camera calibration is known to be inaccurate, feature-based methods are in fact superior. This sensitivity to calibration parameters, while undesirable, could be advantageous as abruptly degraded estimation accuracy using direct methods could indicate sudden mis-calibration errors. Direct algorithms for offline calibration have been previously studied [20, 328, 343, 351, 381], but additional research is needed especially if direct methods are known to be more precise.

- The most common variation of direct methods for visual odometry is Baker and Matthews Inverse Compositional (IC) algorithm [25] due to its efficiency. However, as image warping for visual odometry belongs to the class of 2.5D warps, the derivation of the IC is no longer equivalent to the original formulation of LK. This is because the linearization in the 2.5D case is only valid on the 2D surface of the object and not the 3D volume [26]. Nonetheless, the performance of the IC algorithm seems to be on par or better than the traditional LK formulation and the ESM algorithm [206]. Deeper understanding of the reasons behind the good performance of IC in this context could provide additional insights for developing more accurate algorithms.

- Direct methods for pose estimation have an additional limitation. Namely, the requirement of small inter-frame pixel displacements. In this dissertation, we assumed that large motion-induced displacements can be adequately handled in scale space. But, this is not always the case. The issue of large motions has been previously studied in the optical flow estimation literature where feature-matching constraints were integrated alongside intensity data terms in a variational refinement framework [60]. Pose estimation problems could benefit from a similar formulation to improve robustness and convergence for large inter-frame motions.

- In a similar vein, an emerging technique for pose estimation is to combine direct and feature-based methods in what is sometimes called semi-direct pose estimation [128, 359]. For instance, the frame–frame ego-motion of the camera could be estimated robustly using direct methods, but since feature coordinates remain available a multi-frame refinement step is performed via minimizing the reprojection error using sparse bundle adjustment [128]. While we demonstrated that multi-frame refinement is possible using image data only without the need for correspondences, an interesting research direction is to make use of feature matching constraints alongside the photometric information in a large batch nonlinear optimization problem. In this manner, low frequency (textureless) areas of the image could still be used in motion estimation. Most algorithms for pose estimation ignore low-texture areas in favor of focusing the computational effort on high-texture areas such as corners and edges. However, in many applications most of the image is composed of those low-texture areas. The signal contained in those low-texture areas could provide valuable information for pose estimation, especially when integrated with high-frequency image content. This area of research is only beginning to be explored for RGB-D localization and mapping and demonstrating good results [88].

- In this work, we relied exclusively on feature descriptor costs as a nonparametric means to establish illumination-invariant tracking. However, the limitation of this approach is a narrower basin of convergence, especially when using binary descriptors. Interesting research directions in this area include the integration of both intensity and feature descriptors potentially using a learning-based method, or making use of exposure information using modern machine-vision cameras to improve the reliability of the system. For instance, the algorithms could switch to "robust-mode" given feedback from hardware, or other algorithms that detect degradations in imaging conditions.

- In terms of multi-view refinement for photometric VSLAM, we have only scratched the surface. We relied exclusively on small image patches of fixed size. However, other sophisticated techniques could be used to determine the size and shape of the image region. This could be of immense value to handle foreshortening effects due to perspective projection. Examples include adaptive windows based on image content [132, 193], mid-level segmentation-based primitives such as super pixels [1, 76, 306], or integrating scene constraints from prior knowledge of the environment [77, 78].

- From a theoretical standpoint, direct techniques for pose estimation introduce additional challenges in comparison to their feature-based counterparts. In the feature-based pipeline, minimizing the reprojection error yields the maximum likelihood estimate under relatively moderate assumptions. In fact, there is evidence to support that feature reprojection errors tend to be normally distributed [368], or at the least have zero-mean [23], especially when tracked over a long sequence thereby bringing comfort in the fact that algorithms are optimal, at least theoretically. On the other hand, direct techniques rely for the most part on the image gradients to estimate motion, which are known to be highly correlated [24, 67, 314]. The fact is further more complicated by the asymmetry present in image registration problems where a frame is signaled out as a template [94]. Moreover, it is difficult to establish that photometric errors are normally distributed. In fact, it has been demonstrated that for any motion model in the image plane other than translation, errors are heteroscedastic [55]. Additional research into the statistical properties of minimizing the photometric error could be of immense value, especially for large-scale problems involving not only the camera pose, but also the scene structure. Another interesting line of research to equip direct methods with theoretical guarantees is to design feature descriptors such that their residuals are endowed with a parametric distribution.

- Finally, in this work we have shown the feasibility of the joint motion and structure refinement without relying on geometric feature correspondences, which is useful when feature correspondences cannot be established reliably. An interesting line of future research is developing VSLAM methods that could make use of different sources of visual information. For instance, in addition to geometric and photometric data, VSLAM could benefit from a semantic-level understanding of the scene.

Appendices

APPENDIX A

Computer Code for Local Dense Descriptors Used in this Work

Below, we provide a MATLAB implementation of the descriptors evaluated in Chapter 6

```
function d = raw_intensity(I)
  d{1} = double(I);
end

fucnction d = gc(I)
  d{3} = double(I);
  [d{1},d{2}] = gradient(d{3});
end

function d = laplacian(I)
  d{2} = double(I);
  d{1} = abs(imfilter(d{2}, fspecial(‘laplacian’)));
end

function d = df1(I, ks, s)
  [Ix,Iy] = gradient(double(I));
  ii = Ix > 0; D{4} = ii.*Ix; D{3} = ~ii.*Ix;
  ii = Iy > 0; D{2} = ii.*Iy; D{4} = ~ii.*Iy;
  h = fspecial(‘gaussian’, ks, s);
  for i = 1 : 4
    D{i} = imfilter(D{i}, h);
  end
end
```

```
function d = df2(I, ks, s)
  [G{4},G{3}] = gradient(double(I));
  [G{2},G{1}] = gradient(G{1});
  G{5} = gradient(G{3});
  for i = 5 : -1 : 1
    ii = G{i} > 0; d{2*i} = ii.*G{i}; d{2*i-1} = ~ii.*G{i};
  end
end


function d = bp(I, ks, s)
  C = uint32(census_transform(I));
  h = fspecial(‘gaussian’, ks, s);
  for i = 8 : -1 : 1
    d{i} = imfilter(double(bitshift(bitand(C, 2^(i-1)), -(i-1))), h);
  end
end

function D = census_transform(I)
  I = imfilter(double(I), fspecial(‘gaussian’));
  C = I(2:end-1,2:end-1);
  D(2:end-1,2:end-1) = ...
  (C >= I(1:end-2, 1:end-2)) .* 1  + (C >= I(1:end-2, 2:end-1)) .* 2  + ...
  (C >= I(1:end-2, 3:end  )) .* 4  + (C >= I(2:end-1, 1:end-2)) .* 8  + ...
  (C >= I(2:end-1, 3:end  )) .* 16 + (C >= I(3:end,   1:end-2)) .* 32 + ...
  (C >= I(3:end,   2:end-1)) .* 64 + (C >= I(3:end,   3:end  )) .* 128;
end
```

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[2] K. Aftab and R. Hartley, “Lq-bundle adjustment,” in *Image Processing (ICIP), 2015 IEEE International Conference on*, Sept 2015, pp. 1275–1279.

[3] A. Agarwal, D. Maturana, and S. Scherer, “Visual odometry in smoke occluded environments,” Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-15-07, July 2014.

[4] S. Agarwal, N. Snavely, S. Seitz, and R. Szeliski, “Bundle Adjustment in the Large,” in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin Heidelberg, 2010, vol. 6312, pp. 29–42.

[5] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, “Building rome in a day,” *Commun. ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[6] S. Agarwal, K. Mierle, and Others, “Ceres Solver,” http://ceres-solver.org, 2016.

[7] P. Agouris and T. Schenk, “Automated aerotriangulation using multiple image multipoint matching,” *Photogrammetric Engineering and Remote Sensing*, vol. 62, no. 6, pp. 703–710, 1996.

[8] C. Albl, Z. Kukelova, and T. Pajdla, “R6p-rolling shutter absolute camera pose,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2292–2300.

[9] H. Alismail, “Direct multiple view visual simultaneous localization and mapping,” Pittsburgh, PA, August 2015. [Online]. Available: http://www.cs.cmu.edu/~halismai/halismai_proposal.pdf

[10] H. Alismail and B. Browning, “Direct Disparity Space: An Algorithm for Robust and Real-time Visual Odometry,” Robots Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-14-20, Oct 2014.

[11] H. Alismail, B. Browning, and M. B. Dias, "Evaluating Pose Estimation Methods for Stereo Visual Odometry on Robots," in *the 11th Int'l Conf. on Intelligent Autonomous Systems (IAS-11)*, 2010.

[12] H. Alismail, L. D. Baker, and B. Browning, "Continuous trajectory estimation for 3d slam from actuated lidar," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6096–6101.

[13] H. Alismail, B. Browning, and S. Lucey, "Photometric Bundle Adjustment for Vision-based SLAM," in *Asian Conference on Computer Vision*, 2016.

[14] ——, "Bit-Planes: Dense Subpixel Alignment of Binary Descriptors," *CoRR*, vol. abs/1602.00307, 2016.

[15] ——, "Direct visual odometry using bit-planes," *CoRR*, vol. abs/1604.00990, 2016. [Online]. Available: http://arxiv.org/abs/1604.00990

[16] H. Allende, A. C. Frery, J. Galbiati, and L. Pizarro, "M-estimators with asymmetric influence functions: the distribution case," *Journal of Statistical Computation and Simulation*, vol. 76, no. 11, pp. 941–956, 2006.

[17] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578–589, 2003.

[18] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, and S. Zafeiriou, "Feature-Based Lucas-Kanade and Active Appearance Models," *Image Processing, IEEE Transactions on*, vol. 24, no. 9, pp. 2617–2632, Sept 2015.

[19] E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam, and T. Garaas, "Tracking an RGB-D Camera Using Points and Planes," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, Dec 2013, pp. 51–58.

[20] S. Audet, M. Okutomi, and M. Tanaka, "Direct image alignment of projector-camera systems with planar surfaces." in *CVPR*, 2010, pp. 303–310.

[21] C. Audras, A. Comport, M. Meilland, and P. Rives, "Real-time dense appearance-based SLAM for RGB-D sensors," in *Australasian Conf. on Robotics and Automation*, 2011.

[22] H. Badino, D. Huber, and T. Kanade, "Real-time topometric localization," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 1635–1642.

[23] H. Badino, A. Yamamoto, and T. Kanade, "Visual odometry by multi-frame feature integration," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, 2013, pp. 222–229.

[24] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, 2002.

[25] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[26] S. Baker, R. Gross, and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework: Part 5," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-35, November 2003.

[27] S. Baker, R. Gross, I. Matthews, and T. Ishikawa, "Lucas-Kanade 20 Years On: A Unifying Framework: Part 2," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-01, February 2003.

[28] S. Baker, R. Gross, and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework: Part 4," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-04-14, February 2004.

[29] S. Baker, R. Patil, K. M. Cheung, and I. Matthews, "Lucas-Kanade 20 Years On: Part 5," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-04-64, November 2004.

[30] S. Baker, A. Datta, and T. Kanade, "Parameterizing Homographies," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-06-11, March 2006.

[31] G. A. Baker Jr, "The theory and application of the padé approximant method," in *Advances in Theoretical Physics, Volume 1*, vol. 1, 1965, p. 1.

[32] V. Balntas, L. Tang, and K. Mikolajczyk, "BOLD - Binary Online Learned Descriptor For Efficient Image Matching," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[33] A. Bansal, H. Badino, and D. Huber, "Understanding how camera configuration and environmental conditions affect appearance-based localization," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 800–807.

[34] J. Barron and R. Eagleson, "Motion and structure from time-varying optical flow," in *Vision Interface*. Citeseer, 1995, pp. 104–111.

[35] J. Barron and J. Malik, "Color Constancy, Intrinsic Images, and Shape Estimation," in *Computer Vision ECCV 2012*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7575, pp. 57–70.

[36] A. Bartoli, "Groupwise geometric and photometric direct image registration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 12, pp. 2098–2108, 2008.

[37] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008, similarity Matching in Computer Vision and Multimedia.

[38] P. Beardsley, P. Torr, and A. Zisserman, "3d model acquisition from extended image sequences," in *Computer VisionECCV'96*. Springer, 1996, pp. 683–695.

[39] A. E. Beaton and J. W. Tukey, "The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data," *Technometrics*, vol. 16, no. 2, pp. pp. 147–185, 1974.

[40] S. Benhimane and E. Malis, "Real-time image-based tracking of planes using efficient second-order minimization," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1, Sept 2004, pp. 943–948 vol.1.

[41] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *European conference on computer vision*. Springer, 1992, pp. 237–252.

[42] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," in *European conference on computer vision*. Springer, 1996, pp. 329–342.

[43] M. Black and P. Anandan, "A framework for the robust estimation of optical flow," in *Computer Vision, 1993. Proceedings., Fourth International Conference on*, May 1993, pp. 231–236.

[44] A. Blake and A. Zisserman, *Visual reconstruction*. MIT press Cambridge, 1987, vol. 2.

[45] J.-L. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, 2010.

[46] J.-L. Blanco, F.-A. Moreno, and J. González-Jiménez, "The málaga urban dataset: High-rate stereo and lidars in a realistic urban scenario," *International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014. [Online]. Available: http://www.mrpt.org/MalagaUrbanDataset

[47] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 298–304.

[48] R. F. Boisvert, S. E. Howe, D. K. Kahaner, and J. L. Springmann, *Guide to available mathematical software*. Citeseer, 1990.

[49] J. Borenstein, H. Everett, L. Feng *et al.*, "Where am i? sensors and methods for mobile robot positioning," *University of Michigan*, vol. 119, no. 120, p. 15, 1996.

[50] A. Bosch, A. Zisserman, and X. Muoz, "Image Classification using Random Forests and Ferns," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007.

[51] E. Bostanci, "Is Hamming distance only way for matching binary image feature descriptors?" *Electronics Letters*, vol. 50, no. 11, pp. 806–808, May 2014.

[52] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[53] G. Bradski, "The OpenCV library," 2000.

[54] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. IEEE, 1998, pp. 8–15.

[55] J. Bride and P. Meer, "Registration via direct methods: a statistical approach," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–984–I–989 vol.1.

[56] H. Bristow and S. Lucey, "Regression-based image alignment for general object categories," *CoRR*, vol. abs/1407.1957, 2014. [Online]. Available: http://arxiv.org/abs/1407.1957

[57] ——, "In Defense of Gradient-Based Alignment on Densely Sampled Sparse Features," in *Dense correspondences in computer vision*. Springer, 2014.

[58] M. J. Brooks, W. Chojnacki, D. Gawley, and A. Van Den Hengel, "What value covariance information in estimating vision parameters?" in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 302–308.

[59] R. Brooks and T. Arbel, "The importance of scale when selecting pixels for image registration," in *Computer and Robot Vision, 2007. CRV'07. Fourth Canadian Conference on*. IEEE, 2007, pp. 235–242.

[60] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[61] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping," in *ECCV*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3024, pp. 25–36.

[62] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, "Real-time camera tracking and 3d reconstruction using signed distance functions," in *Robotics: Science and Systems (RSS) Conference 2013*, vol. 9. Robotics: Science and Systems, 2013.

[63] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin Heidelberg, 2010, vol. 6314, pp. 778–792. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15561-1_56

[64] D. R. Canelhas, T. Stoyanov, and A. J. Lilienthal, "Sdf tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3671–3676.

[65] V. C. Cardei, B. Funt, and K. Barnard, “Estimating the scene illumination chromaticity by using a neural network,” *J. Opt. Soc. Am. A*, vol. 19, no. 12, pp. 2374–2386, Dec 2002.

[66] D. Caruso, J. Engel, and D. Cremers, “Large-scale direct SLAM for omnidirectional cameras,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 141–148.

[67] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman, “A content-aware image prior,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 169–176.

[68] H. I. Christensen, “Intelligent home appliances,” in *Robotics Research*. Springer, 2003, pp. 319–327.

[69] W. Churchill and P. Newman, “Practice makes perfect? managing and leveraging visual experiences for lifelong navigation,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4525–4532.

[70] J. Civera, A. J. Davison, and J. M. Montiel, “Inverse depth parametrization for monocular slam,” *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 932–945, 2008.

[71] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel, “Towards semantic slam using a monocular camera,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1277–1284.

[72] D. M. Cole and P. M. Newman, “Using laser range data for 3d slam in outdoor environments,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1556–1563.

[73] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson, “Object recognition and full pose registration from a single image for robotic manipulation,” in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 48–55.

[74] A. Comport, E. Malis, and P. Rives, “Accurate Quadrifocal Tracking for Robust 3D Visual Odometry,” in *Robotics and Automation, 2007 IEEE International Conference on*, April 2007, pp. 40–45.

[75] A. I. Comport, E. Malis, and P. Rives, “Real-time quadrifocal visual odometry,” *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 245–266, 2010.

[76] A. Concha and J. Civera, “Using superpixels in monocular slam,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 365–372.

[77] A. Concha, W. Hussain, L. Montano, and J. Civera, “Manhattan and piecewise-planar constraints for dense monocular mapping,” in *Robotics: Science and systems*, 2014.

[78] ——, “Incorporating scene priors to dense monocular mapping,” *Autonomous Robots*, vol. 39, no. 3, pp. 279–292, 2015.

[79] S. D. Conte and C. W. D. Boor, *Elementary numerical analysis: an algorithmic approach*. McGraw-Hill Higher Education, 1980.

[80] P. Corke, R. Paul, W. Churchill, and P. Newman, "Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 2085–2092.

[81] M. Corsini, M. Dellepiane, F. Ponchio, and R. Scopigno, "Image-to-Geometry Registration: a Mutual Information Method exploiting Illumination-related Geometric Properties," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1755–1764, 2009.

[82] A. Criminisi, I. Reid, and A. Zisserman, "A plane measuring device," *Image and Vision Computing*, vol. 17, no. 8, pp. 625–634, 1999.

[83] A. Crivellaro and V. Lepetit, "Robust 3D Tracking with Descriptor Fields," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[84] A. Crivellaro, P. Fau, and V. Lepetit, "Dense Methods for Image Alignment with an Application to 3D Tracking," EPFL, Tech. Rep. EPFL-REPORT-197866, 2014.

[85] M. Cummins and P. Newman, "Accelerated appearance-only slam," in *Robotics and automation, 2008. ICRA 2008. IEEE international conference on*. IEEE, 2008, pp. 1828–1833.

[86] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 303–312.

[87] S. Daftry, D. Dey, H. Sandhawalia, S. Zeng , J. A. D. Bagnell, and M. Hebert, "Semi-Dense Visual Odometry for Monocular Navigation in Cluttered Environment," in *IEEE International Conference on Robotics and Automation (ICRA) workshop on Recent Advances in Sensing and Actuation for Bioinspired Agile Flight*, May 2015.

[88] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration," *arXiv preprint arXiv:1604.01093*, 2016.

[89] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society Conference on*, vol. 1, June 2005, pp. 886–893 vol. 1.

[90] A. Dame and E. Marchand, "Accurate real-time tracking using mutual information," in *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, Oct 2010, pp. 47–56.

[91] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[92] A. J. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1403–1410.

[93] A. J. Davison and D. W. Murray, “Simultaneous localization and map-building using active vision,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 865–880, 2002.

[94] G. Dedeoglu, T. Kanade, and S. Baker, “The asymmetry of image registration and its application to face tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 807–823, May 2007.

[95] A. Delaunoy and M. Pollefeys, “Photometric bundle adjustment for dense multiview 3d modeling,” in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1486–1493.

[96] F. Dellaert, “Visual SLAM Tutorial: Bundle Adjustment,” 2014.

[97] F. Dellaert and R. Collins, “Fast image-based tracking by selective pixel integration,” in *Proceedings of the ICCV Workshop on Frame-Rate Vision*, 1999, pp. 1–22.

[98] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun, “Structure from motion without correspondence,” in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2. IEEE, 2000, pp. 557–564.

[99] O. Demetz, “Feature invariance versus change estimation in variational motion estimation,” Ph.D. dissertation, UniversitÃd’t des Saarlandes, Postfach 151141, 66041 SaarbrÃijcken, 2015.

[100] D. Demirdjian and T. Darrell, “Motion estimation from disparity images,” in *In Proc. of Int’l Conf. on Computer Vision (ICCV)*, 2001.

[101] R. Deriche and G. Giraudon, “Accurate corner detection: An analytical study,” in *Computer Vision, 1990. Proceedings, Third International Conference on*. IEEE, 1990, pp. 66–70.

[102] M. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (slam) problem,” *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 229–241, 2001.

[103] N. Dowson and R. Bowden, “Mutual Information for Lucas-Kanade Tracking (MILK): An Inverse Compositional Formulation,” *PAMI*, vol. 30, no. 1, pp. 180–185, Jan 2008.

[104] R. O. Dror, E. H. Adelson, and A. S. Willsky, “Estimating surface reflectance properties from images under unknown illumination,” in *In Proc. of SPIE*, vol. 4299, 2001, pp. 231–242.

[105] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.

[106] E. Eade and T. Drummond, “Scalable monocular slam,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 469–476.

[107] J. Engel, J. Stueckler, and D. Cremers, “Large-Scale Direct SLAM with Stereo Cameras,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[108] J. Engel, V. Koltun, and D. Cremers, “Direct Sparse Odometry,” *ArXiv e-prints*, Jul. 2016.

[109] J. Engel, J. Sturm, and D. Cremers, “Semi-Dense Visual Odometry for a Monocular Camera,” in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1449–1456.

[110] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM,” in *ECCV*, 2014.

[111] C. Engels, H. Stewénius, and D. Nister, “Bundle adjustment rules,” in *In Photogrammetric Computer Vision*, 2006.

[112] G. D. Evangelidis and E. Z. Psarakis, “Parametric image alignment using enhanced correlation coefficient maximization,” *PAMI*, vol. 30, no. 10, 2008.

[113] M. F. Fallon, P. Marion, R. Deits, T. Whelan, M. Antone, J. McDonald, and R. Tedrake, “Continuous humanoid locomotion over uneven terrain using stereo fusion,” in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 881–888.

[114] Z. Fang and S. Scherer, “Real-time Onboard 6DoF Localization of an Indoor MAV in Degraded Visual Environments Using a RGB-D Camera,” in *2015 IEEE International Conference on Robotics and Automation*, May 2015.

[115] ——, “"Experimental Study of Odometry Estimation Methods using RGB-D Cameras",” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2014.

[116] Z. Fang, S. Yang, S. Jain, G. Dubey, S. M. Maeta, S. Roth, S. Scherer, Y. Zhang, and S. T. Nuske, “Robust Autonomous Flight in Constrained and Visually Degraded Environments,” in *Field and Service Robotics*, June 2015.

[117] H. Farid and E. P. Simoncelli, “Differentiation of discrete multidimensional signals,” *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 496–508, 2004.

[118] R. Fattal, “Single image dehazing,” in *ACM transactions on graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 72.

[119] O. Faugeras, Q.-T. Luong, and T. Papadopoulo, *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press, 2004.

[120] J.-C. Faugère, G. Moroz, F. Rouillier, and M. S. El Din, "Classification of the perspective-three-point problem, discriminant variety and real solving polynomial systems of inequalities," in *Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*. ACM, 2008, pp. 79–86.

[121] J. Figat, T. Kornuta, and W. Kasprzak, "Performance Evaluation of Binary Descriptors of Local Features," in *Computer Vision and Graphics*. Springer, 2014, pp. 187–194. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-11331-9_23

[122] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," *arXiv preprint arXiv:1504.06852*, 2015.

[123] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, no. 6, 1981.

[124] D. Fleet and Y. Weiss, "Optical Flow Estimation," in *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds. Springer US, 2006, pp. 237–257.

[125] D. Fleet and A. Jepson, "Stability of phase information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 12, pp. 1253–1268, Dec 1993.

[126] A. Flint, C. Mei, I. Reid, and D. Murray, "Growing semantically meaningful models for visual slam," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 467–474.

[127] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," *arXiv preprint arXiv:1506.06825*, 2015.

[128] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," in *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

[129] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 2, pp. 78–90, 2012.

[130] D. Freedman and M. W. Turek, "Illumination-invariant tracking via graph cuts," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 10–17.

[131] R. B. Freeman Jr, "Contrast interpretation of brightness constancy." *Psychological Bulletin*, vol. 67, no. 3, p. 165, 1967.

[132] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," *Foundations and Trendső in Computer Graphics and Vision*, vol. 9, no. 1-2, pp. 1–148, 2015. [Online]. Available: http://dx.doi.org/10.1561/0600000052

[133] Y. Furukawa and J. Ponce, "Accurate camera calibration from multi-view stereo and bundle adjustment," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[134] J. Gallier, "Notes on differential geometry and lie groups," *University of Pennsylvania*, 2012.

[135] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[136] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking," *International Journal of Computer Vision*, vol. 94, no. 3, pp. 335–360, 2011.

[137] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d Reconstruction in Real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[138] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[139] M. A. Gennert and S. Negahdaripour, "Relaxing the brightness constancy assumption in computing optical flow," 1987.

[140] P. Getreuer, "Linear Methods for Image Interpolation," *Image Processing On Line*, vol. 1, 2011.

[141] E. Grafarend and J. Shan, "Closed-form solution of p4p or the three-dimensional resection problem in terms of möbius barycentric coordinates," *Journal of Geodesy*, vol. 71, no. 4, pp. 217–231, 1997.

[142] F. S. Grassia, "Practical parameterization of rotations using the exponential map," *Journal of graphics tools*, vol. 3, no. 3, pp. 29–48, 1998.

[143] K. Grauman and R. Fergus, "Learning binary hash codes for large-scale image search," in *Machine learning for computer vision*. Springer, 2013, pp. 49–87.

[144] A. Gruen, "Adaptive least squares correlation: a powerful image matching technique," *South African Journal of Photogrammetry, Remote Sensing and Cartography*, vol. 14, no. 3, pp. 175–187, 1985.

[145] A. Gruen and T. S. Huang, *Calibration and orientation of cameras in computer vision*. Springer Science & Business Media, 2013, vol. 34.

[146] J. A. Grunert, "Das Pothenostische Problem in erweiterter Gestalt nebst über seine," 1841. [Online]. Available: https://archive.org/details/archivdermathem02grungoog

[147] Y. Guo, "A novel solution to the p4p problem for an uncalibrated camera," *Journal of mathematical imaging and vision*, vol. 45, no. 2, pp. 186–198, 2013.

[148] D. Gutierrez-Gomez, W. Mayol-Cuevas, and J. Guerrero, "Inverse depth for accurate photometric and geometric error minimisation in RGB-D dense visual odometry," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 83–89.

[149] D. Hafner, O. Demetz, and J. Weickert, "Why Is the Census Transform Good for Robust Optic Flow Computation?" in *Scale Space and Variational Methods in Computer Vision*. Springer Berlin Heidelberg, 2013, vol. 7893.

[150] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 10, pp. 1025–1039, Oct 1998.

[151] R. W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[152] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393, 1974.

[153] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011, vol. 114.

[154] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison, "Real-Time Camera Tracking: When is High Frame-Rate Best?" in *European Conf. on Computer Vision (ECCV)*. Springer Berlin Heidelberg, 2012, vol. 7578, pp. 222–235.

[155] K. Hanna, "Direct multi-resolution estimation of ego-motion and structure from motion," in *Visual Motion, 1991., Proceedings of the IEEE Workshop on*. IEEE, 1991, pp. 156–162.

[156] R. Haralick and H. Joo, "2d-3d pose estimation," in *Pattern Recognition, 1988., 9th International Conference on*, Nov 1988, pp. 385–391 vol.1.

[157] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *Int'l Journal of Computer Vision (IJCV)*, 1994.

[158] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[159] C. G. Harris and J. Pike, "3d positional integration from image sequences," *Image and Vision Computing*, vol. 6, no. 2, pp. 87–90, 1988.

[160] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[161] R. Hartley, F. Kahl, C. Olsson, and Y. Seo, "Verifying global minima for l2 minimization problems in multiple view geometry," *International Journal of Computer Vision*, vol. 101, no. 2, 2013.

[162] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *International journal of computer vision*, vol. 103, no. 3, pp. 267–305, 2013.

[163] R. I. Hartley, "Theory and practice of projective rectification," *International Journal of Computer Vision*, vol. 35, no. 2, pp. 115–127, 1999.

[164] H. W. Haussecker and D. J. Fleet, "Computing optical flow with physical models of brightness variation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 661–673, 2001.

[165] J. Heel, "Direct estimation of structure and motion from multiple frames," DTIC Document, Tech. Rep., 1990.

[166] ——, "Dynamic motion vision," in *1989 Advances in Intelligent Robotics Systems Conference*. International Society for Optics and Photonics, 1990, pp. 758–769.

[167] J. Heinly, E. Dunn, and J.-M. Frahm, "Comparative evaluation of binary features," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 759–773.

[168] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition*, 2005.

[169] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics-theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[170] B. Horn, *Robot vision*. MIT press, 1986.

[171] B. K. P. Horn and E. J. Weldon, "Direct methods for recovering motion," pp. 51–76, 1988.

[172] B. K. Horn and B. Schunck, ""Determining optical flow:" a retrospective," *Artif. Intell. v59*, pp. 81–87, 1994.

[173] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1, pp. 185–203, 1981.

[174] Ma, Yi and Soatto, Stefano and Kosecka, Jana and Sastry, S. Shankar, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003.

[175] E. Hsiao and M. Hebert, "Gradient networks: Explicit shape matching without extracting edges," in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[176] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *International Symposium on Robotics Research (ISRR)*, 2011, pp. 1–16.

[177] P. Huber, *Robust Statistics*. New York: Wiley, 1974.

[178] A. Hurlbert, "Colour vision: primary visual cortex shows its influence," *Current Biology*, vol. 13, no. 7, pp. R270–R272, 2003.

[179] D. Q. Huynh, "Metrics for 3D rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.

[180] M. Irani and P. Anandan, "Robust multi-sensor image alignment," in *Computer Vision, 1998. Sixth International Conference on*, Jan 1998, pp. 959–966.

[181] ——, "About Direct Methods," in *Vision Algorithms: Theory and Practice*. Springer Berlin Heidelberg, 2000, pp. 267–277.

[182] M. Irani, P. Anandan, and M. Cohen, "Direct Recovery of Planar-Parallax from Multiple Frames," in *Vision Algorithms: Theory and Practice*, ser. Lecture Notes in Computer Science, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Berlin Heidelberg, 2000, vol. 1883, pp. 85–99.

[183] V. G. Ivancevic and T. T. Ivancevic, "Lecture Notes in Lie Groups," *ArXiv e-prints*, Apr. 2011.

[184] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," in *In Proc. UIST*. ACM Symposium on User Interface Software and Technology, October 2011.

[185] M. Jaimez and J. Gonzalez-Jimenez, "Fast visual odometry for 3-d range sensors," *Robotics, IEEE Transactions on*, vol. 31, no. 4, pp. 809–822, 2015.

[186] Y. Jeong, D. Nister, D. Steedly, R. Szeliski, and I. S. Kweon, "Pushing the envelope of modern methods for bundle adjustment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1605–1617, Aug 2012.

[187] G. Jose and A. Brian, *Precision Landmark Location for Machine Vision and Photogrammetry: Finding and Achieving the Maximum Possible Accuracy*. London: Springer London, 2008. [Online]. Available: http://dx.doi.org/10.1007/978-1-84628-913-2_1

[188] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 4605–4611.

[189] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Trans. on Robotics (TRO)*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[190] M. Kaess, V. Ila, R. Roberts, and F. Dellaert, "The Bayes Tree: An Algorithmic Foundation for Probabilistic Robot Mapping," in *Algorithmic Foundations of Robotics IX*, ser. Springer Tracts in Advanced Robotics, D. Hsu, V. Isler, J.-C. Latombe, and M. Lin, Eds. Springer Berlin Heidelberg, 2011, vol. 68, pp. 157–173.

[191] F. Kahl, S. Agarwal, M. K. Chandraker, D. Kriegman, and S. Belongie, "Practical global optimization for multiview geometry," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 271–284, 2008. [Online]. Available: http://dx.doi.org/10.1007/s11263-007-0117-1

[192] O. Kahler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. H. S. Torr, and D. W. Murray, "Very high frame rate volumetric integration of depth images on mobile device," *IEEE Transactions on Visualization and Computer Graphics Proceedings International Symposium on Mixed and Augmented Reality 2015*, vol. 22, no. 11, 2015.

[193] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE transactions on Pattern analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, 1994.

[194] T. Kanade, P. Rander, and P. Narayanan, "Virtualized reality: Constructing virtual worlds from real scenes," *IEEE multimedia*, no. 1, pp. 34–47, 1997.

[195] Y. Kanazawa and K. Kanatani, "Do we really have to consider covariance matrices for image features?" in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 301–306.

[196] S. Kaneko, I. Murase, and S. Igarashi, "Robust image registration by increment sign correlation," *Pattern Recognition*, vol. 35, no. 10, pp. 2223 – 2234, 2002.

[197] D. Katz, "Die erscheinungsweisen der farben und ihre beeinflussung durch die individuelle erfahrung," 1911.

[198] C. Kerl, J. Sturm, and D. Cremers, "Robust Odometry Estimation for RGB-D Cameras," in *Int'l Conf. on Robotics and Automation (ICRA)*, May 2013.

[199] ——, "Dense visual slam for RGB-D cameras," in *Int'l Conf. on Intelligent Robots and Systems*, 2013.

[200] C. Kerl, J. Stuckler, and D. Cremers, "Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2264–2272.

[201] R. G. Keys, "Cubic convolution interpolation for digital image processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 6, pp. 1153–1160, 1981.

[202] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison, "Simultaneous mosaicing and tracking with an event camera," *J. Solid State Circ*, vol. 43, pp. 566–576, 2008.

[203] J. H. Kim, C. Cadena, and I. Reid, "Direct semi-dense slam for rolling shutter cameras," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1308–1315.

[204] A. A. Kirillov, *An introduction to Lie groups and Lie algebras*. Cambridge University Press Cambridge, 2008, vol. 113.

[205] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, Nov 2007, pp. 225–234.

[206] S. Klose, P. Heise, and A. Knoll, "Efficient compositional approaches for real-time robust direct visual odometry from RGB-D data," in *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2013.

[207] L. Kneip and P. Furgale, "Opengv: A unified and generalized approach to real-time calibrated geometric vision," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1–8.

[208] L. Kneip and H. Li, "Efficient computation of relative pose for multi-camera systems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 446–453.

[209] K. Konolige and M. Agrawal, "FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1066–1077, Oct 2008.

[210] K. Konolige and W. Garage, "Sparse Sparse Bundle Adjustment." in *BMVC*, 2010, pp. 1–11.

[211] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[212] Z. Kukelova and T. Pajdla, "A minimal solution to the autocalibration of radial distortion," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–7.

[213] R. Kumar, P. Anandan, and K. Hanna, "Direct recovery of shape from multiple views: a parallax based approach," in *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1, Oct 1994, pp. 685–688 vol.1.

[214] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna, "Representation of scenes from collections of images," in *Representation of Visual Scenes, 1995.(In Conjuction with ICCV'95), Proceedings IEEE Workshop on*. IEEE, 1995, pp. 10–17.

[215] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Estimating the natural illumination conditions from a single outdoor image," *International Journal of Computer Vision*, 2011.

[216] ——, "Detecting ground shadows in outdoor consumer photographs," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 322–335.

[217] G. H. Lee, B. Li, M. Pollefeys, and F. Fraundorfer, "Minimal solutions for pose estimation of a multi-camera system," in *Robotics Research*. Springer, 2016, pp. 521–538.

[218] P. Y. Lee, "Geometric optimization for computer vision," Ph.D. dissertation, Citeseer, 2005.

[219] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*. Ieee, 1991, pp. 1442–1447.

[220] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.

[221] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2548–2555.

[222] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quart. J. Appl. Maths.*, vol. II, no. 2, pp. 164–168, 1944.

[223] G. Levi and T. Hassner, "LATCH: Learned Arrangements of Three Patch Codes," *CoRR*, vol. abs/1501.03719, 2015. [Online]. Available: http://www.openu.ac.il/home/hassner/projects/LATCH

[224] T. Lindeberg, *Scale-space theory in computer vision*. Springer, 1994.

[225] C. Liu, J. Yuen, and A. Torralba, "SIFT Flow: Dense Correspondence across Scenes and Its Applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/pami/pami33.html#LiuYT11

[226] F. Liu, C. Shen, I. Reid, and A. van den Hengel, "Online unsupervised feature learning for visual tracking," *Image and Vision Computing*, vol. 51, pp. 84–94, 2016.

[227] J. L. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?" in *Advances in Neural Information Processing Systems*, 2014, pp. 1601–1609.

[228] M. Loog and F. Lauze, "The improbability of harris interest points," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 6, pp. 1141–1147, 2010.

[229] C. Loop and Z. Zhang, "Computing rectifying homographies for stereo vision," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 1. IEEE, 1999.

[230] M. Lourakis and A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm," Technical Report 340, Institute of Computer Science-FORTH, Heraklion, Crete, Greece, Tech. Rep., 2004.

[231] M. I. Lourakis and A. A. Argyros, "Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment?" in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1526–1531.

[232] S. Lovegrove, A. Davison, and J. Ibanez-Guzman, “Accurate visual odometry from a rear parking camera,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 788–793.

[233] S. Lovegrove and A. J. Davison, “Real-time spherical mosaicing using whole image alignment,” in *European Conference on Computer Vision*. Springer, 2010, pp. 73–86.

[234] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[235] B. D. Lucas and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA),” in *Proc. of the 1981 DARPA Image Understanding Workshop*, April 1981, pp. 121–130.

[236] S. Lucey, R. Navarathna, A. B. Ashraf, and S. Sridharan, “Fourier Lucas-Kanade Algorithm,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, 2013.

[237] S. P. MacEvoy and M. A. Paradiso, “Lightness constancy in primary visual cortex,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 15, pp. 8827–8831, 2001.

[238] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, “Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles,” in *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, 2014.

[239] K. Madsen, H. B. Nielsen, and O. Tingleff, “Methods for non-linear least squares problems,” Technical University of Denmark, Tech. Rep., 2004.

[240] M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the Mars Exploration Rovers,” *Journal of Field Robotics, Special Issue on Space Robotics*, vol. 24, 2007.

[241] E. Malis, “Improving vision-based control using efficient second-order minimization techniques,” in *Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004 IEEE International Conference on*, vol. 2, April 2004, pp. 1843–1848 Vol.2.

[242] E. Malis and M. Vargas, “Deeper understanding of the homography decomposition for vision-based control,” INRIA, Research Report RR-6303, 2007. [Online]. Available: https://hal.inria.fr/inria-00174036

[243] R. Mandelbaum, G. Salgian, and H. Sawhney, “Correlation-based estimation of egomotion and structure from motion and stereo,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, 1999, pp. 544–550 vol.1.

[244] R. Maronna, D. Martin, and V. Yohai, *Robust statistics*. John Wiley & Sons, Chichester. ISBN, 2006.

[245] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. pp. 431–441, 1963.

[246] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt and Co., Inc., 1982.

[247] S. Martull, M. Peris, and K. Fukui, "Realistic CG stereo image dataset with ground truth disparity maps," in *ICPR workshop TrakMark2012*, vol. 111, 2012, pp. 117–118.

[248] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 810–815, June 2004.

[249] L. Matthies and S. Shafer, "Error Modeling in Stereo Navigation," *IEEE Journal of Robotics and Automation*, 1987.

[250] L. Matthies, T. Kanade, and R. Szeliski, "Kalman Filter-,based Algorithms for Estimating Depth from Image Sequences," *International Journal of Computer Vision*, vol. 5, no. 3, pp. 209 – 236, 1989.

[251] L. Matthies, M. Maimone, A. Johnson, Y. Cheng, R. Willson, C. Villalpando, S. Goldberg, A. Huertas, A. Stein, and A. Angelova, "Computer vision on mars," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 67–92, 2007.

[252] C. Mei and I. Reid, "Modeling and generating complex motion blur for real-time tracking," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[253] C. Mei, S. Benhimane, E. Malis, and P. Rives, "Efficient Homography-Based Tracking and 3-D Reconstruction for Single-Viewpoint Sensors," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1352–1364, Dec 2008.

[254] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, Mar 2002.

[255] M. Meilland and A. Comport, "On unifying key-frame and voxel-based dense visual SLAM at large scales," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 3677–3683.

[256] M. Meilland, A. Comport, and P. Rives, "A spherical robot-centered representation for urban navigation," in *IROS*, Oct 2010.

[257] M. Meilland, P. Rives, and A. I. Comport, "Dense RGB-D mapping for real-time localisation and navigation," in *IV12 Workshop on Navigation Positionnig and Mapping*, Alcalá de Henares, Spain., June 3 2012.

[258] M. Meilland, T. Drummond, and A. I. Comport, "A Unified Rolling Shutter and Motion Blur Model for 3D Visual Registration," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 2016–2023.

[259] S. Meister, B. Jähne, and D. Kondermann, "Outdoor stereo camera system for the generation of real-world benchmark data sets," *Optical Engineering*, vol. 51, no. 02, p. 021107, 2012.

[260] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, Oct 2005.

[261] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, "A Comparison of Affine Region Detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[262] M. Milford, C. Shen, S. Lowry, N. Suenderhauf, S. Shirazi, G. Lin, F. Liu, E. Pepperell, C. Lerma, B. Upcroft, and I. Reid, "Sequence searching with deep-learnt depth for condition- and viewpoint-invariant route-based place recognition," in *6th International Workshop on Computer Vision in Vehicle Technology, in conjunction with IEEE Conference on Computer Vision and Pattern Recognition (CVVT'15)*, 2015.

[263] M. Milford and A. George, "Featureless visual processing for slam in changing outdoor environments," in *Field and Service Robotics*. Springer, 2014, pp. 569–583.

[264] M. J. Milford, G. F. Wyeth, and D. Rasser, "Ratslam: a hippocampal model for simultaneous localization and mapping," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 1. IEEE, 2004, pp. 403–408.

[265] M. Milford and G. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 1643–1649.

[266] M. Mohamed, H. Rashwan, B. Mertsching, M. Garcia, and D. Puig, "Illumination-Robust Optical Flow Using a Local Directional Pattern," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, no. 9, pp. 1499–1508, Sept 2014.

[267] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM review*, vol. 45, no. 1, pp. 3–49, 2003.

[268] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Aaai/iaai*, 2002, pp. 593–598.

[269] H. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," in *tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University and doctoral dissertation, Stanford University*, ser. 1980. CMU/Stanford, September 1980, no. CMU-RI-TR-80-03.

[270] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "User Guide for MINPACK-1," *ANL-80-74, Argonne National Laboratory*, 1980. [Online]. Available: http://www.mcs.anl.gov/~{}more/ANL8074a.pdf

[271] F. Moreno-Noguer, V. Lepetit, and P. Fua, "Accurate non-iterative o (n) solution to the pnp problem," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[272] P. Moulon, P. Monasse, R. Marlet, and Others, "Openmvg. an open multiple view geometry library." https://github.com/openMVG/openMVG.

[273] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3d reconstruction," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 363–370.

[274] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza, "Lifetime estimation of events from dynamic vision sensors," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4874–4881.

[275] M. Muja and D. G. Lowe, "Fast matching of binary features," in *Computer and Robot Vision (CRV), 2012 Ninth Conference on*. IEEE, 2012, pp. 404–410.

[276] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System," *CoRR*, vol. abs/1502.00956, 2015.

[277] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[278] T. Müller, C. Rabe, J. Rannacher, U. Franke, and R. Mester, "Illumination-Robust Dense Optical Flow Using Census Signatures," in *Pattern Recognition*, 2011.

[279] H.-H. Nagel, "On the estimation of optical flow: Relations between different approaches and some new results," *Artificial Intelligence*, vol. 33, no. 3, pp. 299 – 324, 1987.

[280] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Robust visual robot localization across seasons using network flows." in *AAAI*, 2014, pp. 2564–2570.

[281] S. Negahdaripour and B. Horn, "Direct passive navigation: Analytical solution for planes," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, Apr 1986, pp. 1157–1163.

[282] P. Nelson, W. Churchill, I. Posner, and P. Newman, "From dusk till dawn: localisation at night using artificial light sources," in *ICRA*. IEEE, 2015.

[283] A. Netz and M. Osadchy, "Using specular highlights as pose invariant features for 2d-3d pose estimation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 721–728.

[284] P. Neubert, N. Sunderhauf, and P. Protzel, “Appearance change prediction for long-term navigation across seasons,” in *Mobile Robots (ECMR), 2013 European Conference on*. IEEE, 2013, pp. 198–203.

[285] R. Newcombe, S. Lovegrove, and A. Davison, “DTAM: Dense tracking and mapping in real-time,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2320–2327.

[286] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, “KinectFusion: Real-time dense surface mapping and tracking,” in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.

[287] K. Ni, D. Steedly, and F. Dellaert, “Out-of-core bundle adjustment for large-scale 3d reconstruction,” in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[288] D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry,” in *Computer Vision and Pattern Recognition (CVPR)*, June 2004.

[289] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[290] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern Recognition*, vol. 29, pp. 51–59, 1996.

[291] J. Oliensis, “Direct multi-frame structure from motion for hand-held cameras,” in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1, 2000, pp. 889–895 vol.1.

[292] ——, “Multiframe structure from motion in perspective,” in *Representation of Visual Scenes, 1995.(In Conjuction with ICCV’95), Proceedings IEEE Workshop on*. IEEE, 1995, pp. 77–84.

[293] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[294] S. Omari, M. Bloesch, P. Gohl, and R. Siegwart, “Dense visual-inertial navigation system for mobile robots,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 2634–2640.

[295] P. Ondruska, P. Kohli, and S. Izadi, “Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 21, no. 11, pp. 1251–1258, 2015.

[296] S. Osher, “Level set methods,” in *Geometric level set methods in imaging, vision, and graphics*. Springer, 2003, pp. 3–20.

[297] G. Panin and A. Knoll, “Mutual Information-Based 3D Object Tracking,” *International Journal of Computer Vision*, vol. 78, no. 1, pp. 107–118, 2008.

[298] G. Pascoe, W. Maddern, and P. Newman, “Direct visual localisation and calibration for road vehicles in changing city environments,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 9–16.

[299] M. Peris, A. Maki, S. Martull, Y. Ohkawa, and K. Fukui, “Towards a simulation driven stereo vision system,” in *Pattern Recognition (ICPR), 2012 21st International Conference on*, Nov 2012, pp. 1038–1042.

[300] M. Pollefeys, R. Koch, and L. Van Gool, “Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters,” *International Journal of Computer Vision*, vol. 32, no. 1, pp. 7–25, 1999.

[301] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, “Visual modeling with a hand-held camera,” *International Journal of Computer Vision*, vol. 59, no. 3, pp. 207–232, 2004.

[302] J. Ponce, “What is a camera?” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1526–1533.

[303] V. A. Prisacariu, O. Kahler, D. W. Murray, and I. D. Reid, “Real-time 3d tracking and reconstruction on mobile phones,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 21, no. 5, pp. 557–570, 2015.

[304] H. A. Rashwan, M. A. Mohamed, M. A. García, B. Mertsching, and D. Puig, “Illumination Robust Optical Flow Model Based on Histogram of Oriented Gradients,” in *Pattern Recognition*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 8142, pp. 354–363.

[305] I. Reid, “Towards semantic visual SLAM,” in *Control Automation Robotics Vision (ICARCV), 2014 13th International Conference on*, Dec 2014, pp. 1–1.

[306] C. Y. Ren, V. A. Prisacariu, and I. D. Reid, “gSLICr: SLIC superpixels at over 250Hz,” *ArXiv e-prints*, Sep. 2015.

[307] R. Richa and H. Gregory, “Robust Similarity Measures for Gradient-based Direct Visual Tracking,” CIRL, Tech. Rep., 2012.

[308] A. W. Roe, L. Chelazzi, C. E. Connor, B. R. Conway, I. Fujita, J. L. Gallant, H. Lu, and W. Vanduffel, “Toward a unified theory of visual area v4,” *Neuron*, vol. 74, no. 1, pp. 12–29, 2012.

[309] D. M. Rosen, M. Kaess, and J. J. Leonard, “An incremental trust-region method for robust online sparse least-squares estimation,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1262–1269.

[310] P. L. Rosin, “Measuring corner properties,” *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, 1999.

[311] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 430–443.

[312] P. J. Rousseeuw and C. Croux, “Alternatives to the median absolute deviation,” *Journal of the American Statistical association*, vol. 88, no. 424, pp. 1273–1283, 1993.

[313] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2564–2571.

[314] D. L. Ruderman, “The statistics of natural images,” *Network: computation in neural systems*, vol. 5, no. 4, pp. 517–548, 1994.

[315] R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. Davison, “SLAM++: Simultaneous Localisation and Mapping at the Level of Objects,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1352–1359.

[316] R. Salas-Moreno, B. Glocken, P. Kelly, and A. Davison, “Dense planar SLAM,” in *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, Sept 2014, pp. 157–164.

[317] O. Saurer, M. Pollefeys, and G. H. Lee, “A minimal solution to the rolling shutter pose estimation problem,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1328–1334.

[318] H. Sawhney, “3D geometry from planar parallax,” in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR ’94., 1994 IEEE Computer Society Conference on*, Jun 1994, pp. 929–934.

[319] G. G. Scandaroli, M. Meilland, and R. Richa, “Improving NCC-Based Direct Visual Tracking,” in *ECCV*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7577, pp. 442–455.

[320] D. Scaramuzza and F. Fraundorfer, “Visual Odometry [Tutorial],” *Robotics Automation Magazine, IEEE*, vol. 18, pp. 80–92, Dec 2011.

[321] H. Scharr, “Optimal second order derivative filter families for transparent motion estimation,” in *Signal Processing Conference, 2007 15th European*. IEEE, 2007, pp. 302–306.

[322] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[323] C. Schmid, R. Mohr, and C. Bauckhage, “Evaluation of interest point detectors,” *International Journal of computer vision*, vol. 37, no. 2, pp. 151–172, 2000.

[324] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, June 2006, pp. 519–528.

[325] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2012.

[326] L. Sevilla-Lara, D. Sun, E. G. Learned-Miller, and M. J. Black, *Optical Flow Estimation with Channel Constancy*. Springer International Publishing, 2014, pp. 423–438.

[327] Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. M. Seitz, "The visual Turing test for scene reconstruction," in *3D Vision-3DV 2013, 2013 International Conference on*. IEEE, 2013, pp. 25–32.

[328] Y. Sheikh, A. Hakeem, and M. Shah, "On the direct estimation of the fundamental matrix," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–7.

[329] J. Shi and C. Tomasi, "Good features to track," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Jun 1994, pp. 593–600.

[330] M. Shimizu and M. Okutomi, "Precise sub-pixel estimation on area-based matching," in *ICCV*, vol. 1, 2001, pp. 90–97 vol.1.

[331] A. Shterenlikht and N. Alexander, "Levenberg–marquardt vs powells dogleg method for gurson–tvergaard–needleman plasticity model," *Computer Methods in Applied Mechanics and Engineering*, vol. 237, pp. 1–9, 2012.

[332] H.-Y. Shum and R. Szeliski, "Construction of panoramic mosaics with global and local alignment," *International Journal of Computer Vision*, vol. 36, no. 2, pp. 101–130, February 2000, erratum published July 2002, 48(2):151-152.

[333] G. Sibley, C. Mei, I. Reid, and P. Newman, "Adaptive relative bundle adjustment," in *Robotics: science and systems*, vol. 32, 2009, p. 33.

[334] G. Silveira and E. Malis, "Real-time Visual Tracking under Arbitrary Illumination Changes," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–6.

[335] G. Silveira, E. Malis, and P. Rives, "An efficient direct approach to visual SLAM," *IEEE Transactions on Robotics*, 2008.

[336] E. P. Simoncelli, "Distributed Representation and Analysis of Visual Motion," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, Jan 1993.

[337] E. P. Simoncelli and B. A. Olshausen, "Natural Image Statistics and Neural Representation," *Annual Review of Neuroscience*, vol. 24, pp. 1193–1216, 2001.

[338] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the International Conference on Computer Vision*, vol. 2, Oct. 2003, pp. 1470–1477. [Online]. Available: http://www.robots.ox.ac.uk/~vgg

[339] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM transactions on graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 835–846.

[340] I. Sobel and G. Feldman, "A 3x3 Isotropic Gradient Operator for Image Processing," 1968, never published but presented at a talk at the Stanford Artificial Project.

[341] M. Srinivasan, S. Zhang, and N. Bidwell, "Visually mediated odometry in honeybees," *Journal of Experimental Biology*, vol. 200, no. 19, pp. 2513–2522, 1997. [Online]. Available: http://jeb.biologists.org/content/200/19/2513

[342] F. Stein, "Efficient Computation of Optical Flow Using the Census Transform," in *Pattern Recognition*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3175, pp. 79–86.

[343] G. P. Stein, "Tracking from multiple view points: Self-calibration of space and time," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 1. IEEE, 1999.

[344] G. Stein and A. Shashua, "Model-based brightness constraints: on direct estimation of structure and motion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 9, pp. 992–1015, Sep 2000.

[345] F. Steinbrucker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *ICCV Workshops, IEEE Int'l Conf. on Computer Vision*, 2011.

[346] F. Steinbrucker, C. Kerl, D. Cremers, and J. Sturm, "Large-Scale Multi-resolution Surface Reconstruction from RGB-D Sequences," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 3264–3271.

[347] A. Stentz, H. Herman, A. Kelly, E. Meyhofer, G. C. Haynes, D. Stager, B. Zajac, J. A. Bagnell, J. Brindza, C. Dellin *et al.*, "Chimp, the cmu highly intelligent mobile platform," *Journal of Field Robotics*, vol. 32, no. 2, pp. 209–228, 2015.

[348] C. V. Stewart, "Robust Parameter Estimation in Computer Vision," *SIAM Rev.*, vol. 41, no. 3, pp. 513–537, Sep. 1999.

[349] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why Filter?" *Image Vision Comput.*, Feb. 2012.

[350] J. O. Street, R. J. Carroll, and D. Ruppert, "A note on computing robust regression estimates via iteratively reweighted least squares," *The American Statistician*, vol. 42, no. 2, pp. 152–154, 1988.

[351] S. Sugimoto and M. Okutomi, "Camera self calibration based on direct image alignment," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, Nov 2012, pp. 3240–3243.

[352] D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 2432–2439.

[353] N. Sunderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," *Proceedings of Robotics: Science and Systems XII*, 2015.

[354] R. Szeliski and S. B. Kang, "Direct methods for visual scene reconstruction," in *Representation of Visual Scenes, 1995.(In Conjuction with ICCV'95), Proceedings IEEE Workshop on*. IEEE, 1995, pp. 26–33.

[355] N. Sünderhauf, K. Konolige, S. Lacroix, and P. Protzel, "Visual Odometry Using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle," in *Autonome Mobile Systems 2005*, ser. Informatik aktuell, P. Levi, M. Schanz, R. Lafrenz, and V. Avrutin, Eds. Springer Berlin Heidelberg, 2006, pp. 157–163.

[356] V. L. T. Trzcinski, M. Christoudias and P. Fua, "Boosting Binary Keypoint Descriptors," in *Computer Vision and Pattern Recognition*, 2013.

[357] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for handheld 3D sensors," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 5182–5189.

[358] J. Tang, W.-S. Chen, and J. Wang, "A novel linear algorithm for p5p problem," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 628–634, 2008.

[359] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges, "Semi-direct ekf-based monocular visual-inertial odometry," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 6073–6078.

[360] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys, "Live metric 3d reconstruction on mobile phones," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[361] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, "A new approach to vision-aided inertial navigation," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 4161–4168.

[362] J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 311–330, 2002.

[363] P. Thévenaz, T. Blu, and M. Unser, "Image interpolation and resampling," *Handbook of medical imaging, processing and analysis*, pp. 393–420, 2000.

[364] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[365] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[366] E. Tola, C. Strecha, and P. Fua, "Efficient large-scale multi-view stereo for ultra high-resolution image sets," *Machine Vision and Applications*, vol. 23, no. 5, pp. 903–920, 2012.

[367] P. H. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[368] P. H. Torr, A. Zisserman, and S. J. Maybank, "Robust detection of degenerate configurations for the fundamental matrix," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*. IEEE, 1995, pp. 1037–1042.

[369] P. Torr and A. Zisserman, "Feature Based Methods for Structure and Motion Estimation," in *Vision Algorithms: Theory and Practice*. Springer Berlin Heidelberg, 2000, pp. 278–294.

[370] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660.

[371] L. N. Trefethen and D. Bau III, *Numerical linear algebra*. Siam, 1997, vol. 50.

[372] B. Triggs, P. F. Mclauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle Adjustment – A Modern Synthesis," *LNCS*, January 2000.

[373] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Foundations and Trendső in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008. [Online]. Available: http://dx.doi.org/10.1561/0600000017

[374] T. Tykkälä, C. Audras, and A. I. Comport, "Direct iterative closest point for real-time visual odometry," in *ICCV 2011 Workshops*, 2011.

[375] T. Tykkälä, A. I. Comport, J. Kämäräinen, and H. Hartikainen, "Live RGB-D camera tracking for television production studios," *J. Visual Communication and Image Representation*, vol. 25, no. 1, pp. 207–217, 2014.

[376] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, and M. Pantic, "Generic Active Appearance Models Revisited," in *Computer Vision ACCV 2012*, K. Lee, Y. Matsushita, J. Rehg, and Z. Hu, Eds. Springer Berlin Heidelberg, 2013, pp. 650–663.

[377] R. Unnikrishnan and M. Hebert, "Extracting Scale and Illuminant Invariant Regions Through Color," in *17th British Machine Vision Conference*, September 2006.

[378] M. Unser, “Splines: A perfect fit for signal and image processing,” *Signal Processing Magazine, IEEE*, vol. 16, no. 6, pp. 22–38, 1999.

[379] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, “Autonomous driving in urban environments: Boss and the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[380] V. Usenko, J. Engel, J. Stückler, and D. Cremers, “Direct visual-inertial odometry with stereo cameras,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1885–1892.

[381] L. Valgaerts, A. Bruhn, and J. Weickert, “A variational model for the joint recovery of the fundamental matrix and the optical flow,” in *Joint Pattern Recognition Symposium*. Springer, 2008, pp. 314–324.

[382] T. Vaudrey, H. Badino, and S. Gehrig, “Integrating Disparity Images by Incorporating Disparity Rate,” in *Robot Vision*, ser. Lecture Notes in Computer Science, G. Sommer and R. Klette, Eds. Springer Berlin Heidelberg, 2008, vol. 4931, pp. 29–42.

[383] T. Vaudrey, S. Morales, A. Wedel, and R. Klette, “Generalised residual images effect on illumination artifact removal for correspondence algorithms,” *Pattern Recognition*, vol. 44, no. 9, pp. 2034 – 2046, 2011, computer Analysis of Images and Patterns.

[384] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, “Three-Dimensional Scene Flow,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 475 – 480, March 2005.

[385] J. Ventura, C. Arth, and V. Lepetit, “An efficient minimal solution for multi-camera motion,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 747–755.

[386] V. Vineet, O. Miksik, M. Lidegaard, M. Niessner, S. Golodetz, V. Prisacariu, O. Kähler, D. Murray, S. Izadi, P. Peerez, and P. Torr, “Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction,” in *IEEE International Conference on Robotics and Automation (ICRA 2014)*, May 2015, pp. 75–82.

[387] A. Viswanathan, B. R. Pires, and D. Huber, “Vision based robot localization by ground to satellite matching in gps-denied situations,” in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 192–198.

[388] C. Vogel, S. Roth, and K. Schindler, “An Evaluation of Data Costs for Optical Flow,” in *Pattern Recognition*, ser. Lecture Notes in Computer Science, J. Weickert, M. Hein, and B. Schiele, Eds. Springer Berlin Heidelberg, 2013.

[389] G. Vogiatzis and C. Hernández, “Video-based, real-time multi-view stereo,” *Image and Vision Computing*, vol. 29, no. 7, pp. 434 – 441, 2011.

[390] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, “An Improved Algorithm for TV-L1 Optical Flow,” in *Statistical and Geometrical Approaches to Visual Motion Analysis*, 2009, vol. 5604.

[391] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3d motion understanding," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 29–51, 2011.

[392] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," *arXiv preprint arXiv:1602.00134*, 2016.

[393] J. Weickert, M. Welk, and M. Wickert, *L2-Stable Nonstandard Finite Differences for Anisotropic Diffusion*, 2013, pp. 380–391. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38267-3_32

[394] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1385–1392.

[395] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially Extended KinectFusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, Jul 2012.

[396] T. Whelan, H. Johannsson, M. Kaess, J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," in *IEEE Proc. of Intl' Conf. on Robotics and Automation (ICRA)*, May 2013.

[397] B. P. Wrobel, "Facets stereo vision (FAST vision)a new approach to computer stereo vision and to digital photogrammetry," in *ISPRS Intercommission Conf. Fast Processing of Photogrammetric Data*, 1987, pp. 231–258.

[398] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3057–3064.

[399] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba *et al.*, "Sun database: Large-scale scene recognition from abbey to zoo," in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*. IEEE, 2010, pp. 3485–3492.

[400] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Computer Vision - ECCV'94*. Springer, 1994, pp. 151–158.

[401] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research*, vol. 17, pp. 1–32, 2016.

[402] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.

[403] Z. Zhang, "Microsoft Kinect Sensor and Its Effect," *MultiMedia, IEEE*, vol. 19, no. 2, pp. 4–10, Feb 2012.

[404] ——, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and vision Computing*, vol. 15, no. 1, 1997.

[405] L. Zhao, S. Huang, Y. Sun, L. Yan, and G. Dissanayake, "ParallaxBA: bundle adjustment using parallax angle feature parametrization," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 493–516, 2015.

[406] Y. Zheng and L. Kneip, "A direct least-squares solution to the pnp problem with unknown focal length," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[407] M. Z. Zia, L. Nardi, A. Jack, E. Vespa, B. Bodin, P. H. J. Kelly, and A. J. Davison, "Comparative Design Space Exploration of Dense and Semi-Dense SLAM," *CoRR*, vol. abs/1509.04648, 2015. [Online]. Available: http://arxiv.org/abs/1509.04648