Direct Visual-Inertial Odometry using Epipolar Constraints for Land Vehicles

by

Bismaya Sahoo

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2018

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Autonomously operating vehicles are being developed to take over human supervision in applications such as search and rescue, surveillance, exploration and scientific data collection. For a vehicle to operate autonomously, it is important for it to predict its location with respect to its surrounding in order to make decisions about its next movement. Simultaneous Localization and Mapping (SLAM) is a technique that utilizes information from multiple sensors to not only estimate the vehicle's location but also simultaneously build a map of the environment. Substantial research efforts are being devoted to make pose predictions using fewer sensors. Currently, laser scanners, which are expensive, have been used as a primary sensor for environment perception as they measure obstacle distance with good accuracy and generate a point-cloud map of the surrounding. Recently, researchers have used the method of triangulation to generate similar point-cloud maps using only cameras, which are relatively inexpensive. However, point-clouds generated from cameras have an unobservable scale factor. To get an estimate of scale, measurements from an additional sensor such as another camera (stereo configuration), laser scanners, wheel encoders, GPS or IMU, can be used. Wheel encoders are known to suffer from inaccuracies and drifts, using laser scanners is not cost effective, and GPS measurements come with high uncertainty. Therefore, stereo-camera and camera-IMU methods have been topics of constant development for the last decade.

A stereo-camera pair is typically used with a graphics processing unit (GPU) to generate a dense environment reconstruction. The scale is estimated from the pre-calculated base-line (distance between camera centers) measurement. However, when the environment features are far away, the base-line becomes negligible to be effectively used for triangulation and the stereo-configuration reduces to monocular. Moreover, when the environment is texture-less, information from visual measurements only cannot be used. An IMU provides metric measurements but suffers from significant drifts. Hence, in a camera-IMU configuration, an IMU typically is used only for short-durations, i.e. in-between two camera frames. This is desirable as it not only helps to estimate the global scale, but also to give a pose estimate during temporary camera failure. Due to these reasons, a camera-

IMU configuration is being increasingly used in applications such as in Unmanned Aerial Vehicles (UAVs) and Augmented/ Virtual Reality (AR/VR).

This thesis presents a novel method for visual-inertial odometry for land vehicles which is robust to unintended, but unavoidable bumps, encountered when an off-road land vehicle traverses over potholes, speed-bumps or general change in terrain. In contrast to tightly-coupled methods for visual-inertial odometry, the joint visual and inertial residuals is split into two separate steps and the inertial optimization is performed after the direct-visual alignment step. All visual and geometric information encoded in a key-frame are utilized by including the inverse-depth variances in the optimization objective, making this method a direct approach. The primary contribution of this work is the use of epipolar constraints, computed from a direct-image alignment, to correct pose prediction obtained by integrating IMU measurements, while simultaneously building a semi-dense map of the environment in real-time. Through experiments, both indoor and outdoor, it is shown that the proposed method is robust to sudden spikes in inertial measurements while achieving better accuracy than the state-of-the art direct, tightly-coupled visual-inertial fusion method. In the future, the proposed method can be augmented with loop-closure and re-localization to enhance the pose prediction accuracy. Further, semantic segmentation of point-clouds can be useful for applications such as object labeling and generating obstacle-free path.

Acknowledgments

I would like to thank my supervisors Dr. William Melek and Dr. Mohammad Biglarbegian for their constant support and valuable guidance, without which, this work would not have been possible. Thank you for believing in me.

I would also like to thank my friends Kamal Lamichhane, Hemant Surale, Gokhan Gungor and Robin James for their words of encouragement and constructive criticism at various stages of this work. I want to thank Jeff Graansma and Jeremy Reddekopp for their useful insights and kind help while conducting the field experiments.

No words can be enough describe the undying love and blessing of my parents.

Finally, I want to thank The Almighty for giving me the courage to believe in myself and the strength to overcome my hardships.

Dedication

This thesis is dedicated to my sister.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Autonomous vehicles have found applications in areas such as mine exploration, extraterrestrial surface inspection, search and rescue operations, scientific data collection, etc. More recently, there have been efforts to make our day-to-day commute completely autonomous [67] [68]. At present, modern consumer cars use basic forms of autonomy such as driver assist systems, lane keeping systems, advanced braking systems, etc. A fleet of fully autonomous cars will not only make our daily rides safer [19], but also enhance traffic flow, provide a smoother ride experience, remove the need for humans to constantly stay alert. For a land-vehicle to achieve complete autonomy, it has to be equipped with capabilities that enable perception, advanced control and planning, etc. The perception system of an autonomous vehicle should provide good estimates of where the vehicle is located and what is around it, so that control, planning and collision avoidance modules can operate effectively [67].

Simultaneous Localization and Mapping (SLAM) [20] [3] is a probabilistic technique that combines information from multiple sensors to predict both the robot location as well as the landmark locations in the environment. Over the past decade, SLAM has become an established technique for motion estimation of autonomous robots. Variants of SLAM have been developed to estimate the robot-pose as well as features in the environment at the current time step, using the measurements from just the last time step (filtering), or past few time steps (smoothing) [10]. SLAM combines information from various sensors such as cameras, Inertial Measurement Unit (IMU), Global Positioning Systems (GPS), laser scanners, wheel encoders, etc. However, equipping a vehicle with multitude of sensors not only demands increased requirements for real-time data capture but also requires heavy computational requirements for data fusion. Hence, there has been a desire to extract information and make inferences by using fewer sensors.

Laser Scanners are an integral part of an autonomous-car sensor-suite as they generate a point-cloud representation denoting the distance of objects in the environment around the car [52]. However, they are quite expensive. Recently, point-clouds have been generated using conventional cameras [32] which not only encode the distance but also the color/intensity of the object as well, and thus, opening up possibilities of improved semantic segmentation and classification [53].

As cameras encode rich visual information about the environment, researchers have developed techniques for camera-only odometry as well as environment reconstruction [18] [13] [41]. In order to infer depth from cameras, a stereo-configuration (two cameras) can be used [32] [35]. The range of depth estimation depends on the separation between the two camera centers (base-line). More recently, depth inference has also been possible using only a single (monocular) camera by probabilistically refining depth from a video sequence [18]. The ability of monocular systems to build point-cloud maps has from a video stream has limited the utility of stereo cameras for dense mapping [32] and initialization [23]. Moreover, when the object distance is large (e.g, in high altitude flight), the base-line becomes negligible and the stereo-configuration reduces to monocular. Due to this reason, efforts are being made for non-stereo alternatives to infer depth [50]. Alternatively, some cameras are equipped with a depth sensor which produce (Red Green Blue-Depth) RGB-D images. The free availability of depth without the need for a separate estimation technique has enabled development of RGB-D SLAM systems [1] [59] [72]. However, such sensors usually perceive depth using infra-red waves which do not work in presence of sunlight; thus limiting outdoor use.

Estimation of scale is not possible in monocular camera only SLAM, without the use of a metric sensor [14]. In [24], a sonar is used to estimate the ground plane to predict scale.

In [64] prior knowledge about the height of the camera is used to predict scale. Moreover, monocular-camera only SLAM is known to suffer from scale-drift on large trajectories [65]. Further, in the presence of predominant rotational movements, monocular SLAM methods usually fail due to insufficient epipolar stereo-correspondences [11].

Inertial Measurement Units (IMUs) provide both metric information and rotation estimates. However, developing dead-reckoning methods using an IMU as the only sensor, is infeasible as errors in pose estimation quickly accumulate and grow out of bounds. IMUs are cheap and almost always present in modern camera phones. The two sensors, an IMU and a monocular camera, complement each other well by addressing each other's shortcomings [17]; IMU provides the missing scale and rotation information while the camera helps in keeping IMU errors within acceptable bounds. For this reason, camera-IMU fusion techniques have been developed and deployed in applications such as robotics [9] and augmented/virtual reality (AR/VR) [43].

However, monocular visual-inertial fusion techniques have been limited to key-point based methods [28] [56] [57] which build sparse environment maps, and when used with autonomous systems they need to rely on other sensors, such as laser scanners and sonars, to extract useful information about the environment for critical tasks such as navigation [70] [71]. Recently, direct methods [23] have been developed that build richer and more visually informative semi-dense maps in real-time, providing promising prospects for navigation using only visual and inertial sensors. More recently, the so-called tightly-coupled approaches for visual-inertial fusion, developed originally for key-point based methods, have been extended for the direct visual SLAM framework [17]. However, the joint optimization framework, used in the tight-coupled technique, degrades when the measurements from IMU are affected by sudden, unexpected spikes encountered when deployed on a land-vehicle traversing over bumps, pot-holes or general change in terrain. As a land-vehicle is very likely to traverse over uneven terrain, there is a need to develop a visual-inertial technique which predicts reliable pose estimates even in presence of sudden spikes in inertial measurements.

1.1 Problem Statement

The problem statement can be outlined as follows: The measurements obtained at every time step are the camera measurements ($\mathbf{Z}_c$) encoded as pixels and the inertial measurements ($\mathbf{Z}_I := \{a_x, a_y, a_z, g_x, g_y, g_z\}$). The problem is to predict the pose of the camera-IMU setup by utilizing both visual and inertial measurements to infer/refine the depth (encoded in the inverse-depth representation(D_m)), under the following assumptions:

- There is a prior assumption of inverse-depth map; usually randomly initialized.
- The illumination does not change drastically
- All the measurements are temporally in sequence. i.e. measurements $\mathbf{Z}_c, \mathbf{Z}_I$ at time $t > t_0$ do not appear before t_0.
- The scene is predominantly static; i.e. no moving objects other than the vehicle
- The Lambertian assumption is valid. (uniform surface reflectance from all angles) [63]
- There is enough texture in the surface of the environment
- Camera, IMU or Camera-IMU calibration parameters do not change throughout the experiment.

1.2 Contribution

In this thesis, a novel direct semi-tightly coupled visual-inertial fusion technique is presented which is robust in presence of sudden, unintended spikes in IMU measurements experienced when the camera-IMU platform is mounted on a land-vehicle traversing a bumpy terrain. The primary contribution of this thesis is the development of an optimization framework that enforces epipolar constraints to correct pose priors, obtained by integrating noisy IMU measurements, while taking into account geometric misalignment arising due to direct visual optimization. To the best of the author's knowledge, this thesis is the first to handle sudden spikes in IMU measurements in a direct visual-inertial framework.

1.3 Organization

This thesis starts with a discussion of relevant work in Section 2, followed by brief mathematical preliminaries in Section 3. A background on direct state estimation techniques is provided in Section 4, followed by a detailed description of the methodology in Section 5, experiments in Section 6 and results in Section 7. A final conclusion is made along with scope for future work in Section 8.

Chapter 2

Related Work

The proposed approach for visual-inertial data fusion builds upon the existing frameworks for direct monocular visual SLAM. In this chapter, discussion on relevant research starts with vision-only SLAM in Section 2.1 to justify the visual optimization design choices, followed by recent work on visual-inertial SLAM in Section 2.2.

2.1 Monocular-Vision only SLAM

Although stereo-based techniques for visual odometry have existed for quite sometime, MonoSLAM [18] laid the foundation for monocular visual SLAM, where an Extended Kalman Filter (EKF) based algorithm was used to track and map a few key-points. The inverse-depth parametrization was introduced in [13]. The representation of depth in its inverse-depth form made it possible to represent depths of points from unity to infinity. The measurement model, along with its EKF update rule, is almost universally used in visual SLAM techniques.

Parallel Tracking and Mapping (PTAM) [43] introduced the concept of parallelizing tracking and mapping on separate cores on the same CPU, paving way for real-time applications. Dense Tracking and Mapping (DTAM) [1] introduced the concept of "direct-tracking" and built a dense environment reconstruction by utilizing the parallel architecture of a GPU. Since then, [59], [73] and [72] have taken advantage of parallel GPU architecture and 3D point cloud stitching using Iterative Closet Point (ICP) algorithm to achieve impressive results. However, such methods require the use of GPU and depth cameras which make them infeasible for real-time implementation on resource constrained systems.

The work of [6] builds upon [43] to fuse inertial information using a variant of EKF. The tracking accuracy was further improved in [56] and later in [57] by developing a SLAM framework, complete with loop closure and re-localization to achieve long term stability. However, such techniques use key-point descriptors to first isolate a subset of pixels, which not only demand computational overhead but also result in loss of rich visual information by building only a sparse representation of the environment.

Direct Tracking and Mapping introduced in [1] was used in [23] to perform visual SLAM on gradient-rich image regions to generate a much denser environment reconstruction. This approach avoids costly key point computations and generates a denser map in real-time. This approach was further extended to omni-directional [12] and stereo [22] and was later augmented with pose-graph optimization [21] of [46] to show very accurate results. Dense Piecewise Planar Tracking And Mapping (DPPTAM) [16] used the concept of super-pixels [15], to build an even denser map of the environment, under the assumption that neighboring pixels with similar intensity are likely to lie on one plane. Unlike [16], Multi-level Mapping [36] used a K-D tree to generate almost fully dense reconstruction. In contrast, [28] further sparsifies high-gradient pixels by extracting corners to achieve fast tracking while compromising the reconstruction density. The proposed method finds a middle ground and builds upon [21] to achieve real-time results while not sacrificing computational overhead required for dense reconstructions as in [16], [36] or not losing out on the density reconstructed environment as in [28]. However, since the core visual-tracking methodology is similar in all of these approaches, the proposed method can be easily adapted to achieve trade-offs in either direction; to build dense maps or implement faster tracking.

2.2 Visual-Inertial Fusion

Although visual-inertial fusion techniques have been of interest to researchers for over a decade [60] [42] [40], the work of [54] stands out. In this work, a state vector with current and last few poses are augmented with landmark poses in the current field of view and jointly updated using an Extended Kalman Filter (EKF). [74] is an extension of [54] that is twice as fast. The gain in computational speed is a result of efficient representation of the Hessian matrix in its inverse form, such that quick single-precision operations can be performed. This representation has enabled [74] to be deployed in real-time resource constrained embedded systems. In [51] an ensemble of EKFs were used for visual inertial fusion. However, the method relies on a stereo-camera setup for depth estimation. [49] was yet another improvement on [54] where observability of the linearized terms during the EKF update were analyzed and camera-to-IMU parameters were corrected on the fly.

Recently, [62] proposed a method of on-the-fly scale estimation and camera-IMU extrinsic calibration but this method is based on sparse key points. As the number of landmark poses in direct-methods is significantly larger than key-point methods, an equivalent extension of [54], [74] or [62] results in significant computational overhead. In [47], a tightly coupled approach was used to optimize inertial terms with only “key-frame” images in a sliding window non-linear optimization framework to demonstrate superior accuracy over one-step filtering approaches.

In contrast, inertial-aided direct visual methods have been proposed only recently. In [69], a tightly coupled approach for visual-inertial fusion was proposed using factor-graphs [44]. The use of However, a stereo-camera set-up gives reliable depth estimation at the start, as compared to a monocular setup which used random depth initialization. Since the optimization framework requires estimates close to optimal points, a random initialization as done in a monocular framework, makes the technique [69] give incorrect estimates. Moreover, the fundamental assumptions of depth being an independent measurement gets violated in a monocular setting. In [55], a method was proposed to estimate the scale with high accuracy as well as to reduce drifts in previously mapped areas which was later extended [9] for planning and map building for previously unexplored areas. However the method used sparse key-points which becomes computationally expensive when extended for direct methods. An iterated extended Kalman filter based direct-visual inertial fusion scheme was proposed in [7], where image patches were used as descriptors for photometric feedback. However, the method generates a sparse map of the environment.

In [27], a method is described for unifying multiple IMU measurements into a single factor and sparse landmark features in a structureless approach in a factor-graph [44] framework. However, since the method is sparse and includes features only in a key-frame, it does not scale up for direct-methods. [17] describes a method for joint optimization of inertial and visual residuals (tightly-coupled) in real-time. However, it was noticed during experiments that in the presence of sudden spikes in IMU measurements, its performance degrades. Further, random initialization of inverse-depth renders the joint optimization step sub-optimal. Epipolar constraints were exploited for aligning feature points with ground-truth epipolar lines [8]. However, the technique is sparse and relies on extraction of feature correspondences.

In this thesis, a novel visual-inertial technique is presented by formulating epipolar constraints in a direct-image alignment framework, in contrast to sparse formulations such as in [8]. Within the proposed inertial-epipolar optimization technique, each pixel's inverse depth variance is included and accounted for visual misalignment, to correct noisy pose prior obtained from integration of IMU measurements. By isolating inertial terms from a joint framework and performing inertial-epipolar optimization after direct-visual alignment, the proposed method is able to tackle sudden, spurious spikes in IMU measurements. In the experimental section, a comparison is made with the current state-of-the-art direct visual-inertial method [17] to demonstrate the robustness and increased accuracy of the proposed technique, in presence of sudden bumps experienced by the camera-IMU platform, when mounted a moving land-vehicle. Further, due to the increased accuracy in pose-prediction, the proposed method can be used to build a consistent semi-dense map of the environment.

In the next section, a brief description of some preliminary concepts used in visual-inertial state estimation is presented.

Chapter 3

Preliminaries

In this chapter, mathematical preliminaries are presented in brief. In Section 3.1 the camera model is discussed along with its associated distortion models. In Section 3.2, Lie Groups are briefly described which form the backbone of our optimization objective. In Section 3.3, the IMU model used in this work, is discussed. Finally, in section 3.5, the state vector along with all its associated variables is presented.

3.1 Camera Projection Model

The whole geometric process of a point reflecting the light source through the lens until the final capture on the image-sensor is captured via a pin-hole projection model as shown in Figure 3.1.

In the absence of lens distortion, the image formed on the sensor is not exactly of the same size as that of the actual object. The real world object is transformed into a tiny version of itself and encoded in the image sensor. Each element of the sensor array is identifiable by its address. Larger the number of individual sensor elements greater is the discretization. At any particular instant the intensity value captured by such a sensor element is called referred to as a pixel (or picture cell). Each pixel maps to a part of the

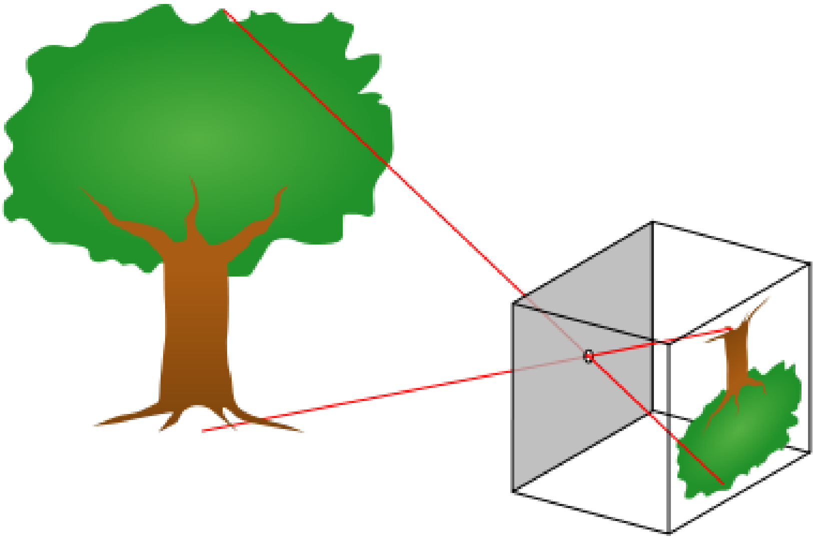


Figure 3.1: Pinhole Camera Model (*Source: Wiki Images*)

actual 3D space which is referred to as 3D points in the rest of this thesis. Hence, it can be said that the camera encodes information about the 3D world in pixels. A relationship of this transformation is given by:

$$\mathbf{x}' = \pi((\mathbf{X})) := \mathbf{K}\mathbf{X} = \begin{pmatrix} f_z & s_y & c_x \\ s_x & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{3.1}$$

where π is called the *Projection Function*, $\mathbf{X}$ is the 3D world coordinate of the pixel, $\mathbf{K}$ is called the *Intrinsic Camera Matrix*, $\mathbf{x}'$ is the pixel coordinate in the image plane, f_x, f_y transform refer to the *focal length parameters* in the x and y directions. c_x, c_y are called the centering parameters, s_x, s_y are the *skew parameters* arising due to misalignment between lens and the sensor. All these parameters are assumed to be fixed and are estimated as a separate calibration step before the experiment.

Alternate to the projection model, a point in the camera image-plane can be unprojected onto the 3D space as:

$$\mathbf{x} = \mathbf{K}^{-1}\mathbf{x}' \tag{3.2}$$

$$with \quad \mathbf{K}^{-1} = \begin{pmatrix} \frac{1}{f_x} & \frac{1}{c_x} & -\frac{c_x}{f_x} \\ \frac{1}{s_y} & \frac{1}{f_y} & -\frac{c_y}{f_y} \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

Hence, if the depth value Z is known, the un-projection model, converts a pixel coordinate to an actual 3D point using:

$$\pi^{-1}(\mathbf{x}, Z) := Z\mathbf{K}^{-1}\mathbf{x} \tag{3.4}$$

Distortion Lens distortion happens because of the spherical lens depending on how much the rays get bent before getting absorbed on the sensor.

Large field of view (FOV) lenses capture larger information about the environment. However, they suffer from distortion [25] [76]. A compact representation of pictorial information complicates spatial association and in-order to represent assign correct spatial coordinates to the pixels, it is essential to undistort the images. The corrected coordinates of pixels suffering from radial distortion is described in [25] [76] and given by :

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + ...) \tag{3.5}$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + ...) \tag{3.6}$$

where x and y are the original pixel coordinates and approximation till the second order suffices in practical cases.

Tangential distortion arises from misalignment of the image sensor axis and the lens axis resulting in a tilt of the image [76]. This kind of distortion is modeled by the equations:

$$x_{corrected} = x + (2p_1xy + p_2(r^2 + 2x^2)) \tag{3.7}$$

$$y_{corrected} = y + (p_1(r^2 + 2x^2) + 2p_2xy) \tag{3.8}$$

In the computer vision community both of these distortion are modeled approximately by the 5 unknown parameters that collectively model the radial distortion.

$$K_{distortion} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad p_3) \tag{3.9}$$

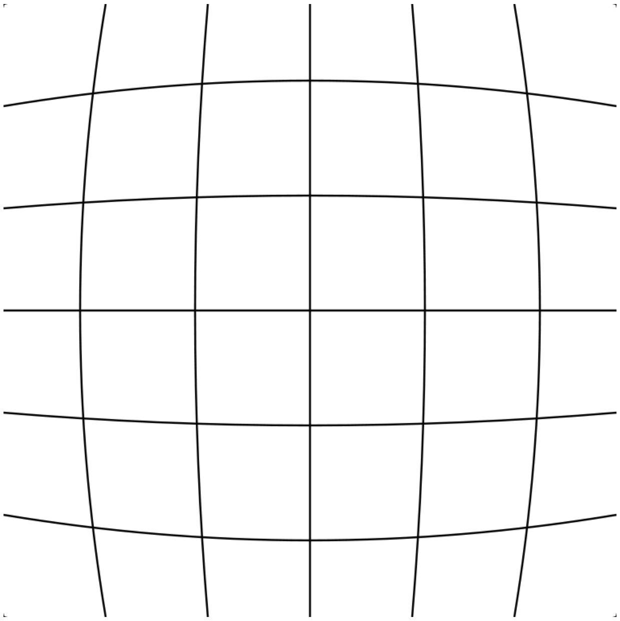


Figure 3.2: An image of an evenly spaced square grid suffering from lens distortion (*Source: Wiki Images*)

3.2 Lie Group and Lie Algebra

The Lie Group $\mathbf{SE(3)}$ is used to represent transformations and poses [27] which encode the rotation as a rotation matrix $\mathbf{R} \in \mathbf{SO(3)}$ and translation $\mathbf{t} \in \mathbb{R}^3$. Lie Algebra is the tangent space to the manifold at identity. The tangent space for the group $\mathbf{SO(3)}$ is denoted by $\mathfrak{so}(3)$ which coincides with the space of 3x3 skew symmetric matrices. Every skew symmetric matrix can be identified with a vector in $\mathbb{R}^3$ using the *hat* operator, $(.)^{\wedge}$:

$$\omega^{\wedge} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathfrak{so}(3) \tag{3.10}$$

Similarly, a skew symmetric matrix is mapped to a vector in $\mathbb{R}^3$ using the *vee* operator $(\cdot)^{\vee}$: for a skew symmetric matrix $S = \omega^{\wedge}$, the vee operator is such that $S^{\vee} = \omega$.

The *exponential map* at identity $exp : \mathfrak{so}(3) \to SO(3)$ associates elements of the Lie Algebra to a rotation:

$$exp(\phi^{\wedge}) = \mathbf{I} + \frac{\sin(||\phi||)}{||\phi||}\phi^{\wedge} + \frac{1-\cos(||\phi||)}{||\phi||^2}(\phi^{\wedge})^2 \tag{3.11}$$

The *logarithm map* (at identity) associates a matrix $\mathbf{R} \in \mathbf{SO(3)}$ to a skew symmetric matrix:

$$\log(\mathbf{R}) = \frac{\varphi \cdot (\mathbf{R} - \mathbf{R}^T)}{2\sin(\varphi)} \text{ with } \varphi = \cos^{-1}\left(\frac{\mathrm{tr}(\mathbf{R}) - 1}{2}\right) \tag{3.12}$$

It is also worthwhile to note that $\log(\mathbf{R})^{\vee}{=}\mathbf{a}\varphi$, where $\mathbf{a}$ and φ are the rotation axes and the rotation angle of $\mathbf{R}$. The mapping is depicted in Figure 3.3.

The use lie algebra in optimization allows for smooth pose updates which obey the properties of manifold operations.

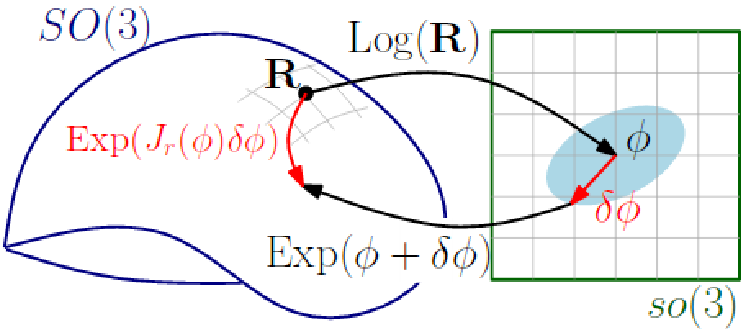


Figure 3.3: Lie group manifold operations (Source [27], ©2017 IEEE)

3.3 IMU Model

Let $\mathbf{R}_j^w \in \mathbf{SO(3)}$ represent the rotation, $\mathbf{t}_j^w \in \mathbb{R}^3$ denote the translation vector and $\mathbf{v}_j^w \in \mathbb{R}^3$ denote the velocity vector in the current frame j in the world reference frame w. This is calculated from the previous frame i by forward Euler integration [17];

$$\mathbf{R}_j^w = \mathbf{R}_i^w \mathbf{R}_j^i \tag{3.13}$$

$$\mathbf{v}_j^w = \mathbf{v}_i^w + \mathbf{v}_{ij}^w \tag{3.14}$$

$$\mathbf{t}_j^w = \mathbf{t}_i^w + \mathbf{t}_{ij}^w \tag{3.15}$$

where $\mathbf{R}_j^i$ denotes the relative rotations between frames i and j, $\mathbf{v}_{ij}^w$ is the incremental velocity and $\mathbf{t}_{ij}^w$ is the translation vector. These variables are computed from the IMU measurements angular velocity, $\boldsymbol{\omega}$, and linear acceleration, $\mathbf{a}$, with biases $\mathbf{b}_\omega$ and $\mathbf{b}_a$ respectively. The increments can be written in terms of measurements as:

$$\mathbf{R}_j^i = \prod_{p=k}^{k+N-1} \exp_{SO(3)}([\boldsymbol{\omega}(p) + \mathbf{b}_\omega(p)]^\wedge \delta t) \tag{3.16}$$

$$\mathbf{v}_{ij}^w = \sum_{p=k}^{k+N-1} (\mathbf{R}_p^w(\mathbf{a}(p) + \mathbf{b}_a(p)) - \mathbf{g})\delta t \tag{3.17}$$

$$\begin{aligned}\mathbf{t}_{ij}^w =& N\mathbf{v}_i^w \delta t + \frac{1}{2}\sum_{p=k}^{k+N-1} (2(k+N-1-p)+1) \\ & (\mathbf{R}_p^w(\mathbf{a}(p) + \mathbf{b}_a(p)) - \mathbf{g})\delta t^2\end{aligned} \tag{3.18}$$

where p denotes the instances where IMU measurements are available in between two camera frames i and j. The IMU biases are modeled as random walk processes with variances $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_\omega$:

$$\mathbf{b}_a(k+1) = \mathbf{b}_a(k) + \boldsymbol{\eta}_a \delta t \tag{3.19}$$

$$\mathbf{b}_\omega(k+1) = \mathbf{b}_\omega(k) + \boldsymbol{\eta}_\omega \delta t \tag{3.20}$$

3.4 Gravity Alignment

In order to obtain a correctly oriented world-frame map, a gravity alignment operation is performed as an initialization step. A few IMU acceleration samples were recorded to estimate the initial World frame to Body Frame orientation (${}^W\mathbf{R}_B \in \mathbf{SO(3)}$). First, the magnitude of the gravity vector is computed as:

$$|g| = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{3.21}$$

The initial pitch and roll angles are then computed as:

$$pitch = \tan^{-1}\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \tag{3.22}$$

$$roll = \tan^{-1}\left(\frac{-a_y}{a_z}\right) \tag{3.23}$$

where a_x, a_y, a_z are the averaged accelerations over the first few frames in the x, y, z Cartesian directions. As the yaw is undetermined from the accelerations alone, initial yaw is assumed to be zero. In the presence of a magnetometer, better initialization to the yaw angle can be performed.

3.5 The State Vector

To aid in the estimation process, a state vector maintains the pose estimates, the updates on velocity and bias estimates. More specifically, the state is defined as $\mathbf{s}_i := [\mathbf{T}_i^T \quad \mathbf{v}_i^T \quad \mathbf{b}_i^T] \in \mathbb{R}^{15}$ where, $\mathbf{b} \in \mathbb{R}^6$ is a vector containing the bias in the 3D acceleration and 3D angular velocity measurements of the IMU. The pose element, $\mathbf{T_i} \in \mathbf{SE(3)}$, encodes the translation and $\mathbf{R}_i \in \mathbf{SO(3)}$ and $\mathbf{t}_i \in \mathbb{R}^3$.

The state-vector, in this work, does not maintain the past states or feature positions unlike feature-based fusion methods due to the following reasons: 1) number of points in dense optimization methods is much more than feature-based methods and including them in the state adds significantly to the size and computational cost and a filtering based approach is adopted over the smoothing approach [29]. The prior-pose estimate to our optimization is obtained by forward Euler integration described in Section 3.3. Once an estimate of the pose $\mathbf{T}_j$ is obtained using method described in Section 5.2, the state vector is updated and used again for the next time step.

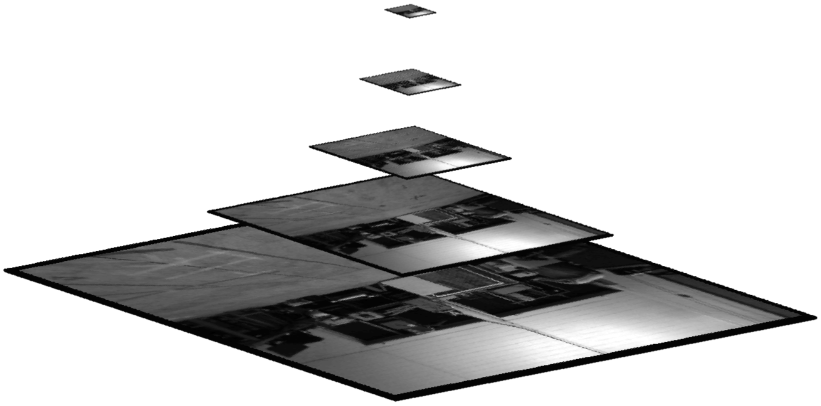


Figure 3.4: Image pyramids shown with the least-resolution, lowest level pyramid at the top to the highest resolution at the bottom. The optimization starts the top and moves down after convergence at a particular level.

3.6 Image Pyramid

To avoid local minima, the optimization is performed over image pyramid (See Figure. 3.4). The bottom-most level of the pyramid is the resolution obtained from the camera sensor (640x480, in our case). The next level is constructed by averaging out four neighboring pixels. The least resolution image is at the top. The optimization starts at the top and gradually moves down the pyramid upon convergence at that level. Finer features in the image are averaged out gradually as one moves up the pyramid. As the optimization objective is usually highly non-linear, such a pyramidal implementation avoids local minimas when optimization is performed from top to bottom [75]. In this work, 5 levels of image-pyramids were used.

Chapter 4

Direct Tracking

This chapter provides a brief background on direct visual tracking methods, upon which the proposed technique is based on.

4.1 Lucas-Kannade Image Alignment

This technique seeks to minimize the photometric residual with an objective function defined as:

$$min \sum_{x} (I_1(\omega(\mathbf{x}, \mathbf{p})) - I_0(\mathbf{x}))^2 \tag{4.1}$$

where $\mathbf{x}$ is the coordinate of a pixel in the template image I_0, $w(.)$ is a warp function that maps the pixel $\mathbf{x}$ to its corresponding location in the target image I_1. The goal of the optimization is to seek optimal parameters $\mathbf{p}$ such that the cost, defined in (4.1), is minimized for a small patch of pixels around the original pixel, $\mathbf{x}$. The vector $\mathbf{p}$ represents the "warp parameters" encoding a transformation of the image patch in I_0 to I_1.

4.2 Direct Image Alignment

This technique is a variant of the Lucas Kannade algorithm [4], where the warp function encodes unprojection of the pixel with an inverse depth D_i (reciprocal of depth) and reprojected back on to the target image, after applying transformation, $\mathbf{T}$, to the unprojected point. More recently, this method has been applied to visual odometry applications [21] [16] [28], yielding impressive results. Here, instead of computing feature points, all pixels with a valid depth estimate (belonging to an inverse-depth map D_m) and having enough intensity gradient are included in a single objective function and the sum of squared intensity residuals is minimized.

$$\begin{aligned} & min \sum (\mathbf{r}_{ph})^2 \\ = & \quad min \sum_{x \in Dm} (I_1(\omega(\mathbf{x}, \mathbf{T}, \mathbf{K})) - I_0(\mathbf{x}))^2 \end{aligned} \tag{4.2}$$

where $\mathbf{T} \in \mathbf{SE(3)}$ is the transformation encoding the rotation $\mathbf{R} \in \mathbf{SO(3)}$ and translation $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{K}$ is the camera calibration matrix.

The formulation of the warp parameters as members of Lie Group allows for smooth updates in the tangent space $\mathfrak{se}(3)$. The minimum is calculated using variants of Gauss-Newton algorithm with increments $\Delta\xi$ as:

$$\Delta\xi = -(\mathbf{J^T W J})^{-\mathbf{1}} \mathbf{J^T W r}_{ph} \tag{4.3}$$

where $\mathbf{J}$ is the stacked Jacobian of all pixels for the residual $\mathbf{r}_{ph}$ with respect to the six elements of Lie Algebra $\Delta\xi$.

It is worth noting here that the relationship between the optimization variables to the cost function (4.2) is highly non-linear due to mathematical operations such a matrix multiplication of the rotation matrix and homogenization in the projection model. Moreover, a random inverse depth map is used to bootstrap the monocular slam process. Although, a robust weighing function [38] is typically deployed to handle outliers, improper initialization of initial transformation estimate, $\mathbf{T}$, can force the optimization to a local minima.

4.3 Visual Inertial Direct Odometry

This technique adds an IMU to aid in the optimization process. The best results have been achieved by 'tightly-coupled' approach, where the photometric residual (4.2) is jointly optimized along with an inertial-residual given by:

$$\mathbf{r}_{imu} = \begin{bmatrix} log_{SO(3)}\Big((\mathbf{R}_j^i)^T(\mathbf{R}_i^w)^T\mathbf{R}_j^w\Big)^\vee \\ \mathbf{t}_j^w - \mathbf{t}_{ij}^w - \mathbf{t}_i^w \\ \mathbf{v}_j^w - \mathbf{v}_{ij}^w - \mathbf{v}_i^w \\ \mathbf{b}_{a_j}^w - \mathbf{b}_{a_i}^w \\ \mathbf{b}_{g_j}^w - \mathbf{b}_{g_i}^w \end{bmatrix} \tag{4.4}$$

where $log_{SO(3)}(.)^\vee$ denotes the retracted rotation residual from Lie Group $\mathbf{SO(3)}$ to Lie algebra $\mathfrak{so}(3)$, $(.)_j^i$ is obtained by integrating IMU measurements from time frame i to j. $(.)^w$ denotes world frame of reference. $(.)_i^w$ is the state at the previous time frame i and $(.)_j^w$ is the parameter to be optimized. $\mathbf{R}, \mathbf{t}, \mathbf{v}, \mathbf{b_a}, \mathbf{b_g}$ denote the rotation, translation, linear velocity, accelerometer bias and gyroscope bias, respectively.

Notice that the residual $\mathbf{r}_{imu}$ is minimized when the predicted state parameters match with the ones obtained from IMU measurements. In a joint estimation framework, where both $\mathbf{r}_{ph}$ and $\mathbf{r}_{imu}$ are minimized simultaneously, the updates in IMU pose, $\Delta\xi_{imu}$, is calculated first and used to "guide" the optimization of the photometric residual. This is typically desirable as the measurements from IMU provide both the initial estimate (by distorting the cost function to generate a new minima around the minima of the IMU residual) and a direction for convergence (through Jacobian of IMU residual with respect to photometric updates $\Delta\xi_{ph}$). However, in presence of unexpected but unavoidable bumps, the new minima for this residual is highly offsetted from where it was desired. Since the original cost function (4.2) is highly non-linear, such an offset makes it susceptible to local minima.

In the next chapter, this issue is addressed by the proposed technique.

Chapter 5

Methodology

In this chapter, the proposed method is described in detail. First, in Section 5.1, the image alignment algorithm, used for visual-alignment, is briefly touched upon. In Section 5.2, a detailed description of the visual-inertial tracking method is presented. Finally, the mapping technique is described in Section 5.3.

5.1 Inverse Compositional Image Alignment

As the first step, the visual residual, as formulated in Section 4.2, is minimized. Although, there are several variants available [4] to minimize the objective function 4.2, the "Inverse Compositional Method" was used in this thesis because of its faster convergence rate. The algorithm is outlined in brief below. The reader is advised to refer to [4] for details.

Pre-compute:

1. Evaluate the gradient ∇I_0 of the template $I_0(\mathbf{x})$
2. Pre-compute the Jacobian $\frac{\partial \omega}{\partial \xi}$ at $(\mathbf{x}; \mathbf{0})$
3. Compute the steepest descent images $\nabla I_0 \frac{\partial \omega}{\partial \xi}$

4. Compute the Hessian matrix using the Equation:

$$H = \sum_x \left[\nabla I_0 \frac{\partial \omega}{\partial \xi}\right]^T \left[\nabla I_0 \frac{\partial \omega}{\partial \xi}\right] \tag{5.1}$$

Iterate:

1. Warp I with $\omega(\mathbf{x}; \mathbf{T})$ to compute $I(\omega(\mathbf{x}; \mathbf{T}))$
2. Compute the error image $I(\omega(\mathbf{x}; \mathbf{T})) - I_0(\mathbf{x})$
3. Compute $\sum_x \left[\nabla I_0 \frac{\partial \omega}{\partial \xi}\right]^T [I(\omega(\mathbf{x}; \mathbf{p})) - I_0(\mathbf{x})]$
4. Compute $\Delta\xi$ using Equation:

$$\Delta\xi = H^{-1} \sum_x \left[\nabla I_0 \frac{\partial \omega}{\partial \xi}\right]^T [I(\omega(\mathbf{x}; \mathbf{T})) - I_0(\mathbf{x})] \tag{5.2}$$

5. Update warp $\omega(\mathbf{x}; \mathbf{T}) \leftarrow \omega(\mathbf{x}; \mathbf{T}) \circ \omega(\mathbf{x};\text{Exp}(\Delta\xi))^{-1}$ until $||\Delta\xi|| \leq \epsilon$

Once converged, the epipolar alignment is performed as described in the next section.

5.2 Visual-Inertial Epipolar Constrained Odometry

In this novel formulation, the IMU residuals were decoupled from the direct visual image alignment step (4.2), where the photometric cost function was allowed to converge with respect to the randomly initialized unscaled inverse depth map. After convergence, all the corresponding points on the target image were not perfectly aligned (but only a subset, the ones which satisfy the brightness consistency asssumption). For the sake of simplicity, the assumption that the optimization yeilds perfect matches ($\mathbf{x_{LK}}$) is made. This assumption later relaxed in Section 5.2.1.

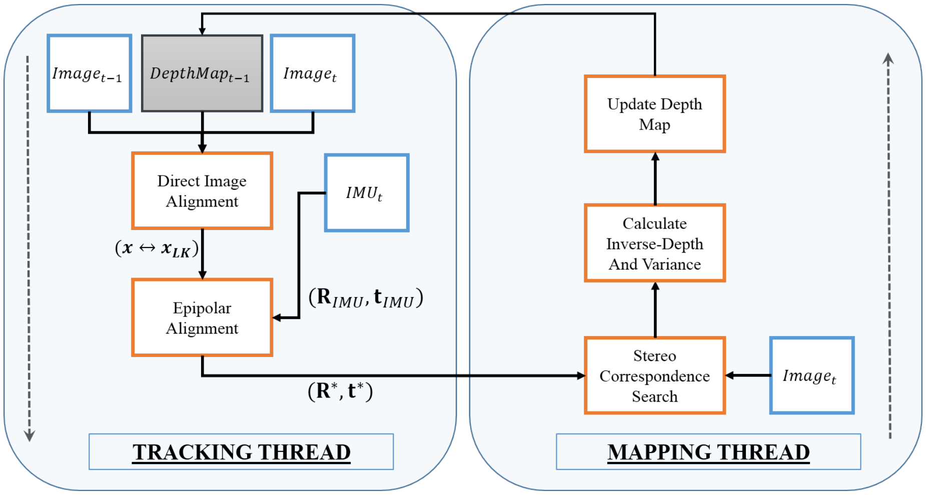

$Image_{t-1}$
$DepthMap_{t-1}$
$Image_t$
Direct Image Alignment
IMU_t
$(x \leftrightarrow x_{LK})$
Epipolar Alignment
$(\mathbf{R}_{IMU}, \mathbf{t}_{IMU})$
$(\mathbf{R}^*, \mathbf{t}^*)$
TRACKING THREAD
Update Depth Map
Calculate Inverse-Depth And Variance
Stereo Correspondence Search
$Image_t$
MAPPING THREAD

Figure 5.1: Schematic for Visual Inertial Epipolar Constrained Odometry. Two threads run in parallel. The tracking thread encodes the epipolar optimization and the mapping thread uses the optimized pose $(\mathbf{R}^*, \mathbf{t}^*)$ to update the map.

Using the prior transformation (described in Section 3.5) $(\hat{\mathbf{T}}^i_{j,IMU})$, for each pixel in the key-frame image, an initial estimate can be computed for the epipolar line, $(\mathbf{l}'^*)$ through the relation:

$$\begin{aligned} \hat{\mathbf{l}}' =& \hat{\mathbf{F}}_{\mathbf{IMU}}\mathbf{x} && (5.3) \\ =& \mathbf{K}^{-\mathbf{T}}[\hat{\mathbf{t}}_{\mathbf{IMU}}]_{\times}\hat{\mathbf{R}}_{\mathbf{IMU}}\mathbf{K}^{-\mathbf{1}}\mathbf{x} && (5.4) \end{aligned}$$

where $\hat{\mathbf{F}}_{\mathbf{IMU}}$ is the initial estimated guess for the Fundamental Matrix constructed through $\hat{\mathbf{T}}^i_{j,IMU} \in \mathbf{SE(3)}$ which encodes $\hat{\mathbf{R}}_{\mathbf{IMU}} \in \mathbf{SO(3)}$ and $\hat{\mathbf{t}}_{\mathbf{IMU}} \in \mathbb{R}^3$, and $\mathbf{x} \in \mathbb{R}^3$ is the homogenized pixel coordinate $(u, v, 1)$. Using (5.4), the epipolar residual is defined as,

$$r_{epl} = dist(\mathbf{x}^{\mathbf{T}}_{\mathbf{LK}}, \hat{\mathbf{l}}') \qquad (5.5)$$

where, $dist(\mathbf{p}, \mathbf{l})$ is the function computing the euclidean distance between point, $\mathbf{p}$ and line, $\mathbf{l}$.

The epipolar constraint dictates that the best match pixel $(\mathbf{x_{bm}})$ correspondig to the source pixel ($\mathbf{x}$) must lie on the corresponding epipolar line ($\mathbf{l'^*}$). The aim was to obtain the optimal transformation ($\mathbf{T}^*$) by applying updates $\Delta\xi \in \mathfrak{se}(3)$ to $\hat{\mathbf{T}}^i_{j,IMU}$(obtained by integrating IMU measurements), such that the perpendicular distance between the epipolar line ($\hat{\mathbf{l}}'$) and $\mathbf{x_{LK}}$ (correspinding point on target image after convergence of (4.2)) on the 2D image plane is minimized. This step is referred to as "epipolar image alignment" in the rest of the paper. The objective was: 1) to find $\mathbf{l'^*}$ so that 1D stereo-search along this line would give $\mathbf{x_{bm}}$ and 2) to obtain $\mathbf{T}^*$ as a result of this alignment.

Note however, that this alignment is 2D and the rank of Fundamental Matrix is 2 which causes a loss of the scale information. This phenomenon can be imagined as "zooming in/out" on a scene where there is perfect 2D epipolar alignment but absence of scale information makes estimation of "zooming in/out" motion impossible.

To address this shortcoming, we formulate an inverse-depth residual to counter any scale drift during the epipolar optimization. The initial scale was first estimated by obtaining a coarse estimate of inverse depths for all pixels $\mathbf{x_{LK}}$ due to the transformation $\hat{\mathbf{T}}^i_{j},_{IMU}$. This was done by finding $\mathbf{x_{LK\perp}}$, which was the perpendicular projection of $\mathbf{x_{LK}}$ on $\hat{\mathbf{l}}'$. The ratio of mean of all such inverse depths to the mean of our initial inverse depth assumption gives a good initial scale estimate. This process of finding scale is inspired by [21] where mean inverse depth is conserved to unity at each key-frame to prevent scale drift.

To conserve the scale drift during epipolar alignment, the following residual is formulated.

$$\begin{aligned} r_{D_i} =& \hat{D}_i - g(\hat{D}_i, \mathbf{T}) && (5.6) \\ =& \hat{D}_i - (\mathbf{R}_{row3} \bullet \mathbf{Kx} + t_z) && (5.7) \end{aligned}$$

where, $\hat{D}_i$ is the initial estimate of the inverse depth of pixel i obtained as explained above, $\mathbf{R}_{row3}$ is the third row of the current Rotation Estimate, ($\bullet$) denotes dot product and t_z is the current translation estimate in 'z' direction.

Our complete cost function becomes:

$$
\begin{aligned}
min \sum (r_{epl}^2 + r_D^2) & \qquad (5.8)\\
= \sum dist(\mathbf{x_{LK}^T}, \hat{\mathbf{l}}')^2 + (\hat{D}_i - g(\hat{D}_i, \mathbf{T}))^2 & \qquad (5.9)\\
= \sum dist(\mathbf{x_{LK}^T}, \mathbf{K}^{-\mathbf{T}}[\mathbf{t}]_\times \mathbf{R}\mathbf{K}^{-\mathbf{1}}\mathbf{x})^2 \quad + & \qquad (5.10)\\
(\hat{D}_i - (\mathbf{R}_{row3} \bullet \mathbf{K}\mathbf{x} + t_z))^2 &
\end{aligned}
$$

where $dist(\mathbf{p}, \mathbf{l})$ is the function computing the euclidean distance between point, $\mathbf{p}$ and line, $\mathbf{l}$.

At this point, one might observe that the inverse depth residual is zero at the start and progressively grows with iteration. This effect is desirable and intended to counter the scale drift as it becomes more and more prominent during minimization of epipolar residual. The image of the epipole of the second image on the template (key-frame) image during the optimization process is shown in Figure 5.2.

5.2.1 Robust Weighting

Earlier in Section 5.2, perfect alignment of pixels as a result of the visual-inertial optimization step was assumed. However, due to random initialization at the start and general noise in camera pixel measurements, this assumption is not valid. In-fact, only a subset of these pixels 'align' themselves well. The extent of alignment dictates the extent of relaxation allowed for a particular pixel in the epipolar alignment step (Section 5.2). In this section, the extent of this alignment is modeled by normalizing the epipolar and the inverse depth residuals (5.10). In addition, robust weighting function is employed to counter the effect of outliers.

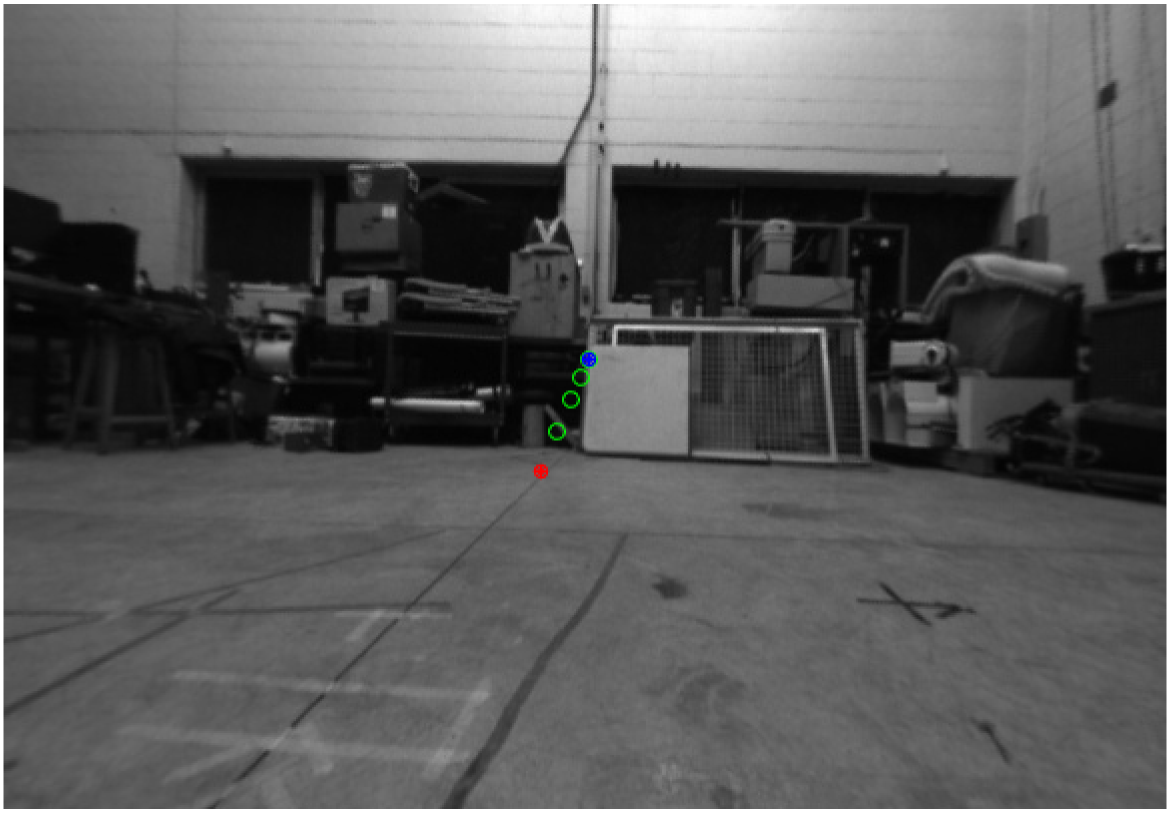


Figure 5.2: The epipole positions plotted on the key-frame image during an optimization process for a straight line motion. RED shows the epipole position due to noisy prior due to integration of IMU measurements at the start of the optimization. GREEN shows intermediate epipole positions during the optimization. BLUE is the final epipole position. Since the trajectory is straight, the epipole's image on the key-frame image should be at the center of the image, which is where the initial noisy pose prior is driven to, as a result of optimization.

Epipolar residuals for each pixel are normalized as follows:

$$r\hat{_{epl}} = \frac{r_{epl}}{\sigma_{r_{epl}}} \tag{5.11}$$

$$\sigma^2_{r_{epl}} = (\frac{\partial r_{epl}}{\partial D})\sigma^2_D + \sigma^2_c \tag{5.12}$$

where σ^2_D is the inverse depth variance, $(\frac{\partial r_{epl}}{\partial D})$ is the Jacobian of the epipolar residual with respect to the inverse depth and σ_c^2 is the camera pixel noise.

Similarly, inverse depth residual is normalized

$$\hat{r_D} = \frac{r_D}{\sigma_{r_D}} \tag{5.13}$$

$$\sigma_{r_D}^2 = (\frac{\partial r_D}{\partial D})\sigma_D^2 \tag{5.14}$$

where $(\frac{\partial r_D}{\partial D})$ is the Jacobian of the inverse depth residual with respect to the inverse depth.

A single Huber weighing function is applied to both the residuals considering the fact that if one pixel is an outlier, both the residuals must be weighted less.

$$w_x := \rho(\hat{r_D}^2 + \hat{r_{epl}}^2) \tag{5.15}$$

$$\rho(r^2) := \begin{cases} 1 & if|r| < \delta \\ \frac{\delta}{|r|} & otherwise \end{cases} \tag{5.16}$$

The complete algorithm can be summarized below:

Initialize:

1. Use the IMU measurements from the last time-step of the state vector, $\mathbf{s_{i-1}}$, (See Section 3.5) to predict pose $\mathbf{T_{init}} \in \mathbf{SE(3)}$ by Euler forward integration as described in Section 3.3.

Iterate:

1. Calculate the epipolar residual, r_{epl} (5.5) and the inverse depth residual r_{D_i} (5.6) using the current estimate of pose, $\mathbf{T}$ for all valid pixels

2. Compute the respective Jacobians of r_{epl} and r_{D_i} w.r.t. pose updates, $\Delta\xi \in \mathfrak{se}(3)$ (See Appendix B and C) for all valid pixels

3. Compute Jacobians of r_{epl} (B.10) and r_{D_i} (C.4) w.r.t. respective inverse depths and compute weights for all valid pixels

4. Compute the Hessian $\mathbf{H} = \mathbf{J^T W J}$

5. Compute the pose update $\Delta\xi = \mathbf{H^{-1}} \sum_x \mathbf{J^T} w(r_D + r_{epl})$

6. Update pose $\mathbf{T_{it}} \leftarrow \mathrm{Exp}(\hat{\Delta\xi}) \circ \mathbf{T_{it-1}}$, where $\circ$ denotes pose composition in $\mathbf{SE(3)}$ and **it** denotes the iteration number.

5.3 Mapping

To enable real-time operation, both the Tracking and Mapping modules are implemented in parallel threads.The mapping thread is blocked until the image is first tracked and has a valid pose ($\mathbf{T_i}^*$), as depicted in Figure 5.1. Each valid pixel is transformed in the key-frame" image (static for comparison with incoming image sequences and the image to which inverse-depth map is assigned) on to a corresponding pixel in the successive "reference" image (each incoming image) and perform a one-dimensional search along five equidistant points along the epipolar lines in both images (See Figure 5.3). Each successful stereo-match is at the point where the Sum of Squared Difference (SSD) error is minimum (See Figure 5.4) and corresponds to the best estimate of the original pixel in the key-frame image (shown as box in Figure 5.3) on corresponding reference image.

We follow a similar methodology to [23] and employ geometric and photometric errors in the stereo computations as briefly described below. The reader is encouraged to refer to [23] for details.

$$\sigma_d^2 := \alpha^2(\sigma^2_{\lambda,photometric} + \sigma^2_{\lambda,geometric}) \tag{5.17}$$

with;

$$\sigma^2_{\lambda,photometric} = \frac{2\sigma_i^2}{g_p^2} \tag{5.18}$$

$$\sigma^2_{\lambda,geometric} = \frac{\sigma_l^2}{\langle g, l\rangle^2} \tag{5.19}$$

where

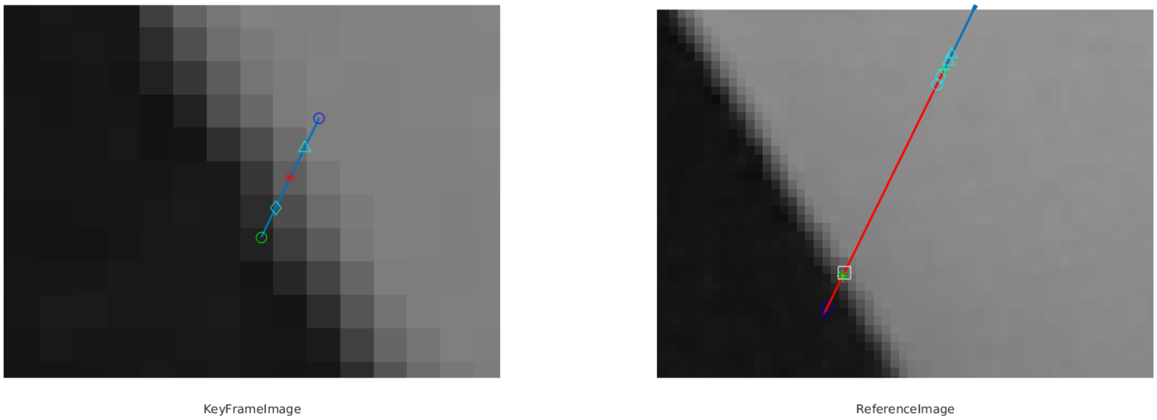


Figure 5.3: Epipolar stereo matching on key-frame and reference images. On the left are five equidistant points on the key-frame image and on the right, are the same five points being searched along the epipolar line (shown as RED line). The best match point, $\mathbf{x_{bm}}$ is shown as a box. The cost associated as this search is performed is shown in Figure 5.4

σ_i: camera-pixel noise
σ_l: variance of positioning error of the initial point on epipolar line
g_p: gradient along the epipolar line
g : normalized image gradient

Following each successful stereo observation, the depth and variance is updated as:

$$\mathcal{N}\left(\frac{\sigma_p^2 d_o + \sigma_o^2 d_p}{\sigma_p^2 + \sigma_o^2}, \frac{\sigma_p^2 \sigma_o^2}{\sigma_p^2 + \sigma_o^2}\right) \tag{5.20}$$

where $\mathcal{N}(d_p, \sigma_p^2)$ is the prior distribution and $\mathcal{N}(d_o, \sigma_o^2)$ is the observed distribution.

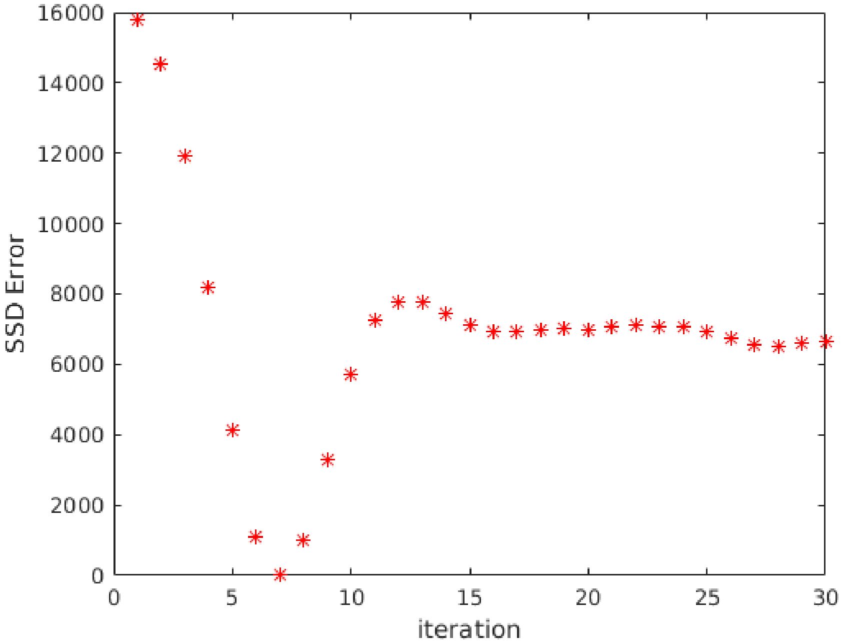


Figure 5.4: Sum of Squared Difference Error as five equidistant points are checked along the epipolar line in the reference image as shown in 5.3 The minima is the point of best match..

Chapter 6

Experimental Hardware Calibration

This chapter of the thesis describes the experiments. We start with the description of the experimental setup in Section 6.1, followed by detailed description of the calibration procedure in Section 6.2.

6.1 Setup

The experimental setup consist of a monocular camera (PointGrey BlackFly @50fps) with a wide FOV lens (90°) to capture 640x480 monochrome images, and IMU (Microstrain 3DGX2 @100Hz) to capture 6-dof linear accelerations and angular velocity. Both of these sensors were rigidly fixed on a base as shown in Figure 6.1. The processor used was a Lenovo Z40 laptop equipped with intel i5 processor and 4GB of RAM, running Ubuntu Linux pre-loaded with Robot Operating System (ROS). Additionally, a Vicon Motion Capture System was used as Ground Truth for indoor experiments.

To highlight the advantages of the proposed method, it was essential to have a set-up that could impart sudden unintentional bumps during movement. To realize that, a makeshift trolley-cart with one misaligned wheel, was used to impart sudden bumps to the set-up. Moreover, the inability to intentionally control the timing, duration or nature of

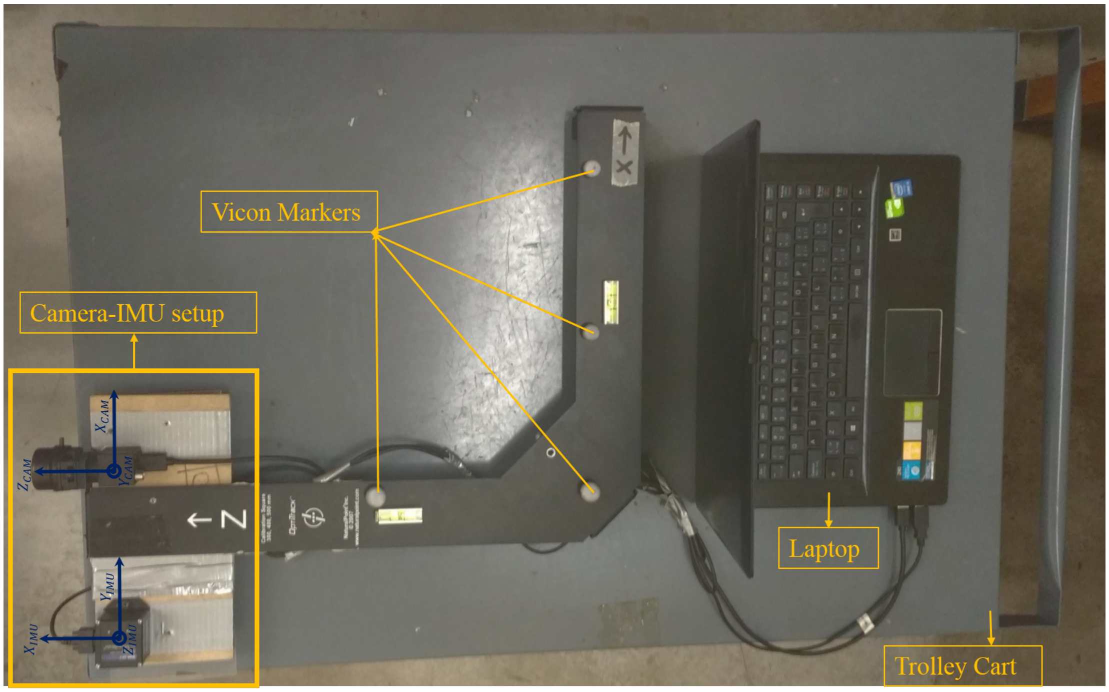


Figure 6.1: Indoor experiment setup. The monocular camera and IMU fixed rigidly and mounted on a trolley-cart with one misaligned wheel.

sudden spikes in IMU measurements, made our system mimic real world outdoor conditions where land-vehicles would encounter sudden bumps or change in terrain.

For outdoor experiments, the camera-IMU platform was mounted at the front-end of an off-road 6x6 vehicle, manufactured by ARGO, as shown in Figure 6.2.

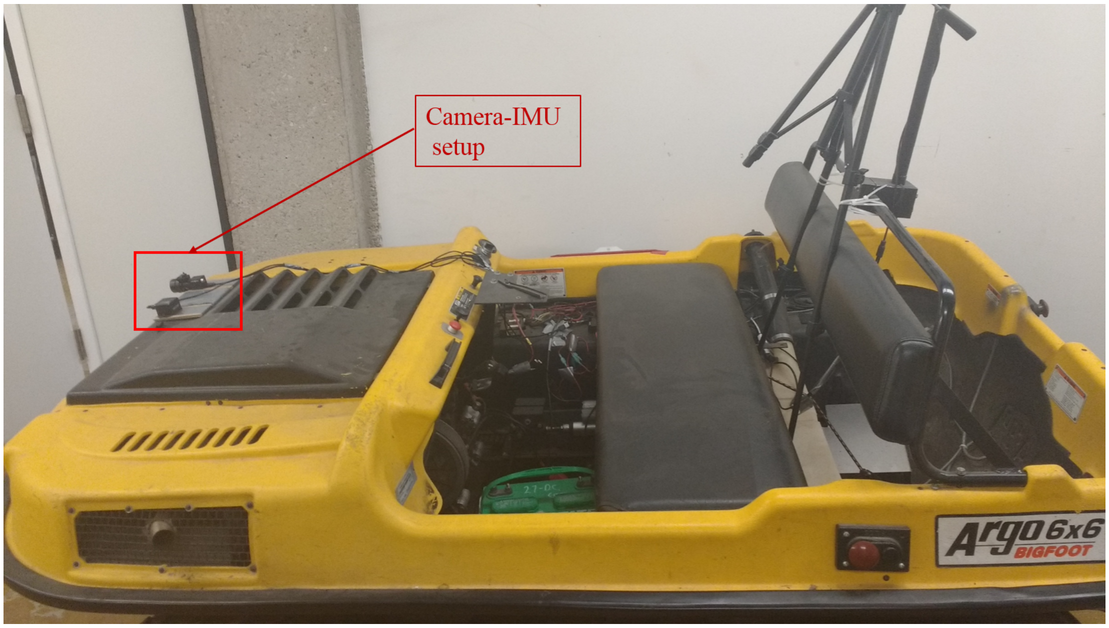


Figure 6.2: Outdoor experiment setup. The monocular camera and IMU fixed rigidly and mounted on an off-road vehicle. The axis conventions are shown in Figure 6.3

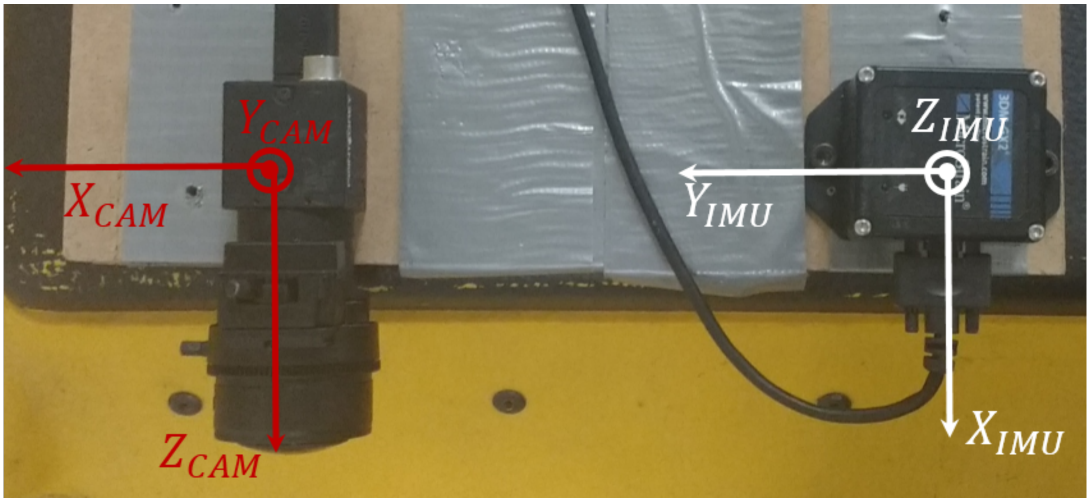


Figure 6.3: Camera-IMU setup close-up view. Axis conventions shown for clarity.

6.2 Hardware Calibration

Before the start of the experiment, the camera-IMU system was calibrated offline in order to determine the focal lengths (f_x, f_y) and camera center in pixels (c_x, c_y), radial distortion parameters, the IMU-variances (using Allan Variance Analysis), IMU biases (b_a, b_g), camera-IMU transformation matrix (T_{ci}) and temporal offsets (time lag between each apparently overlapping camera and IMU sample (See Figure. 6.4). The open-source package $Kalibr$ [31] [30] was used to perform this calibration. Since a large FOV camera was used for the experiments, the radial distortion due to lens was corrected for each incoming image using the distortion model available in the open-source undistorter package inside PTAM [43]. The equations described in Section 5 assume a pre-rectified image, free from radial distortion. The calibration parameters for our camera are shown in Table 6.1:

Table 6.1: Table summarizing calibration parameters for experiments.

PARAMETER :	VALUE
IMU Variances:	0.01(m/s^2) and 0.005(rad/sec)
Temporal Offset:	0.002 sec (See Figure 6.4)
Accelerometer Biases:	b_{a_x} : 0.132, b_{a_y} : 0.015, b_{a_z} : 0.002
Gyroscope Biases:	b_{g_x} : -0.00022, b_{g_y} : -0.00107, b_{g_z} : 0.00042
(f_x, f_y, c_x, c_y) :	$369.70, 367.81, 332.67, 248.46$
Radial Distortion Coeff:	$-0.04, -0.017, 0.033, -0.019$

6.2.1 IMU Calibration

Static calibration of IMU primarily refers to the alignment of both the sensors in all the 3 directions. For the accelerometer, it was first essential to estimate the gravity vector **g**. This value was calculated as the norm of all three measurements at rest and found to be $-9.80556435 m/s^2$.

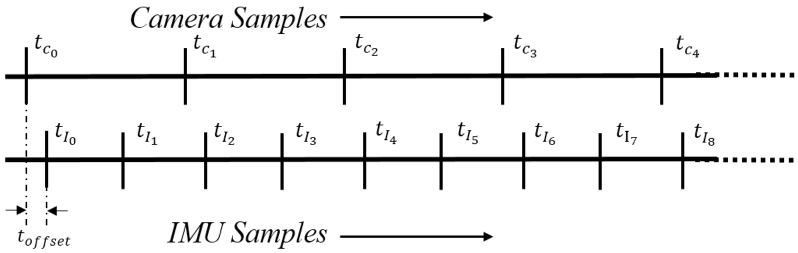


Figure 6.4: Temporal offset between Camera and IMU sampling.

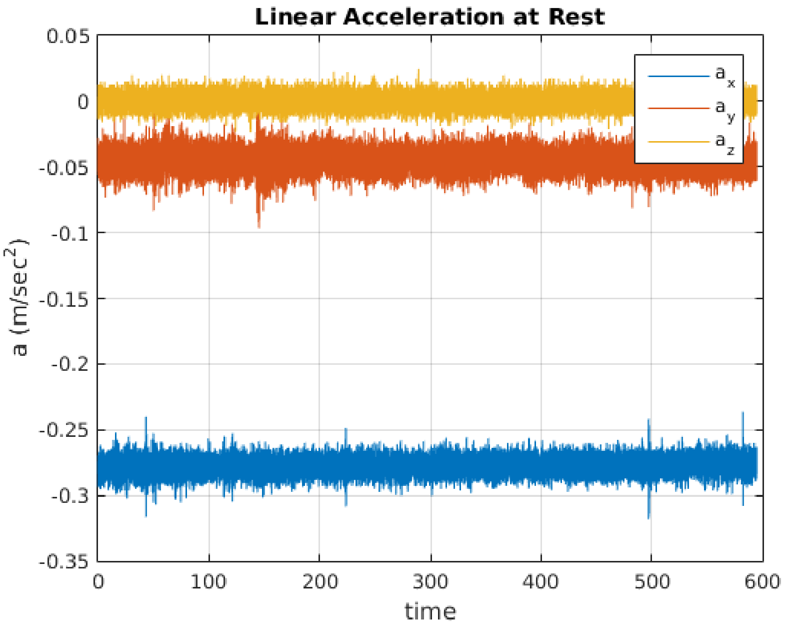


Figure 6.5: Linear acceleration at rest

Accelerometer Calibration

Note that in Figure 6.5 the gravity vector has been subtracted to obtain the normalized accelerations in all the 3 directions at rest. They are not perfectly aligned because of in-exact coincidence with the actual gravity vector. Although, it might seem erroneous at first glance, it is actually advantageous as, one can obtain the *"orientation"* of the sensor. This has been calculated and shown in Figure 6.6. Note that Yaw Pitch and Roll will coincide only when the base of the IMU is perfectly horizontal. This is almost never the case, and we take advantage of the misalignment to estimate the *initial* orientation of the IMU.

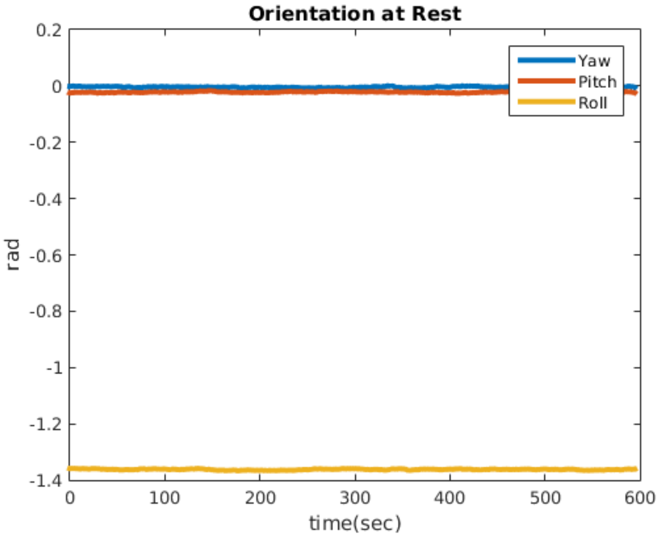


Figure 6.6: Orientation of the IMU at rest calculated from the misalignment with the absolute gravity vector

Gyroscope Calibration

However, for the gyroscope, there was no physical signal correspondence to follow. Hence, gyroscope data was calibrated to remove the drift and bias. At rest, the data should coincide with the zero mean at rest. The calibrated angular velocity plot is shown in Figure 6.7

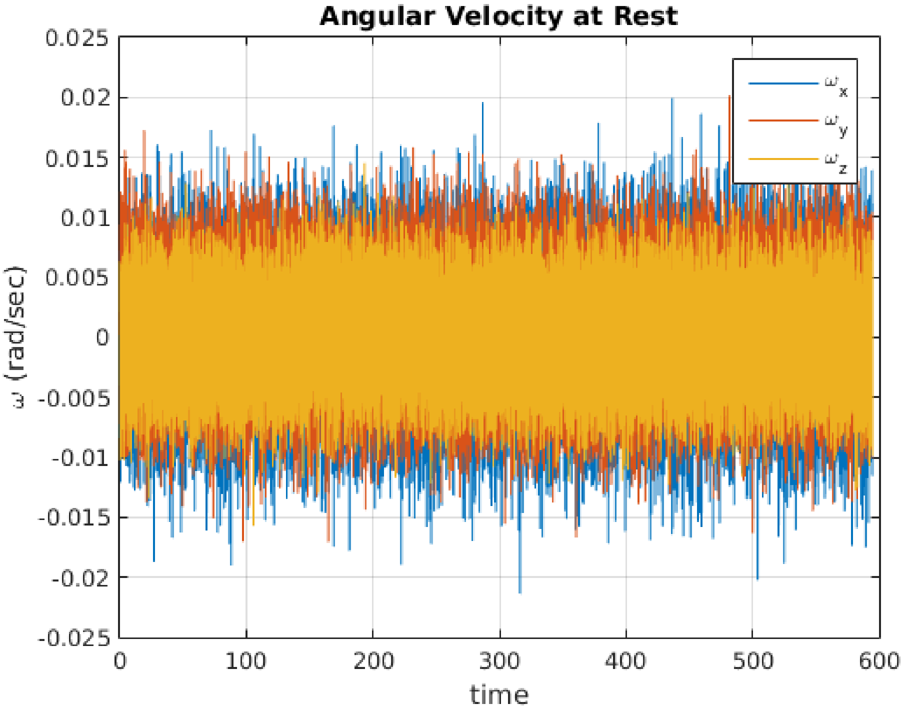


Figure 6.7: Calibrated angular velocity plot

Allan Variance Analysis

Allan Variance Analysis(AVA) was useful to determine the noise parameters of the IMU. To compute the plots a 30 minute stream of IMU data was captured at rest. AVA for our experiment are shown below:

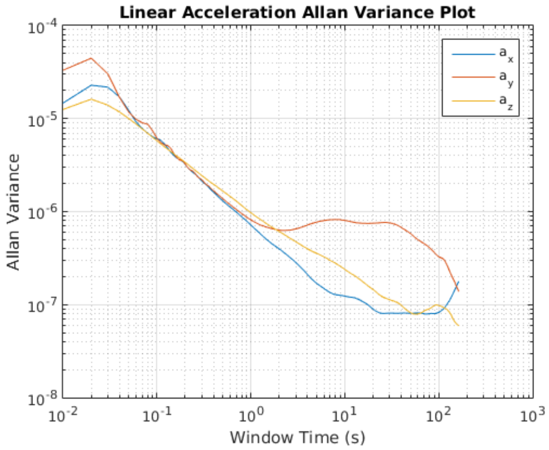


Figure 6.8: Allan variance plot for the linear acceleration in 3 directions

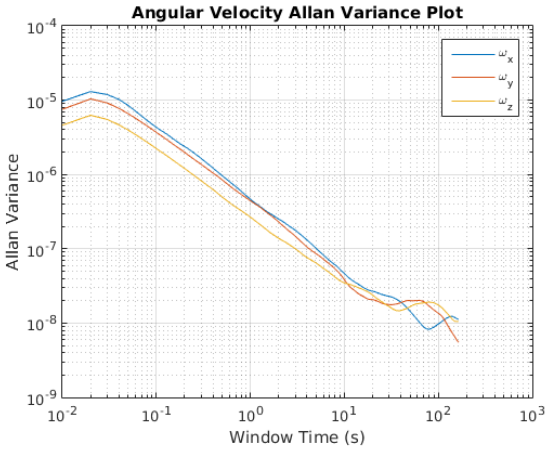


Figure 6.9: Allan variance plot of the gyroscope data in 3 directions

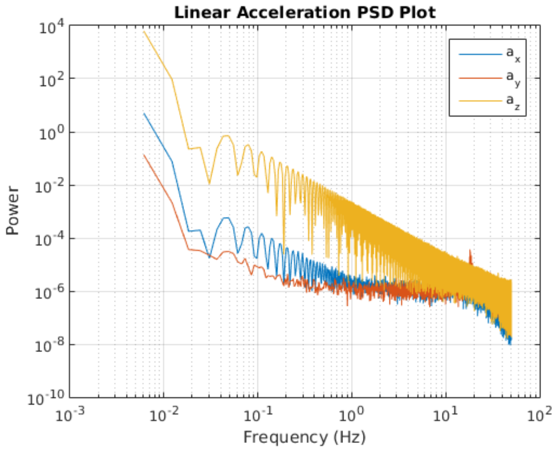


Figure 6.10: Power spectral density plots of the linear acceleration data

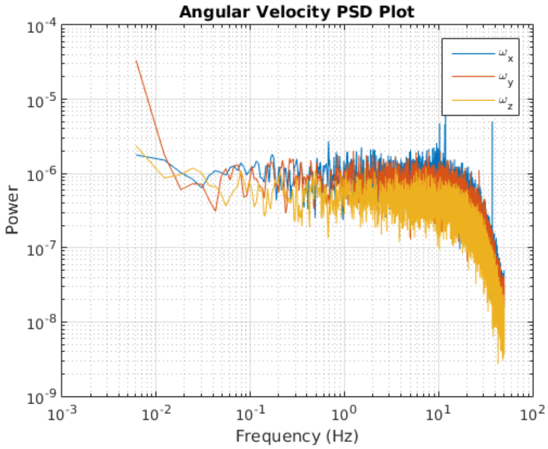


Figure 6.11: Power spectral density plots of the gyroscope data

6.2.2 Camera Calibration

To estimate geometrical relationships between the environment and camera measurements, known 2D/3D correspondences were matched with pixel-pixel distance measurements and scale factors in each direction were established. After such a procedure, the "Camera Model" could be used as f_x, f_y, c_x, c_y which remain fixed during the whole procedure.

To carry-out this calibration, a suitable calibration pattern with known point-point correspondences was needed. Out of all the available options such as checkerboard and circle-board, etc., the April-Tag calibration pattern was chosen as shown in Figure.6.12.

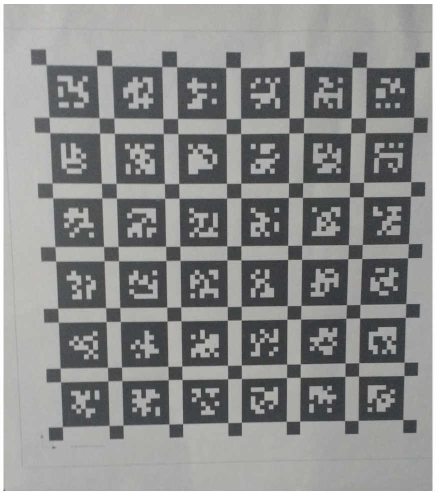


Figure 6.12: April-Tag calibration pattern used for static camera calibration

The bigger the pattern, the better the precision. To achieve this, we used a calibration patter of 80cm X 80cm. Figure 6.13 shows the Seimens star patterns used to check the focus of the cameras.

To estimate the accuracy of the calibration it is necessary to statistically evaluate the results. The results of static calibration are listed below:

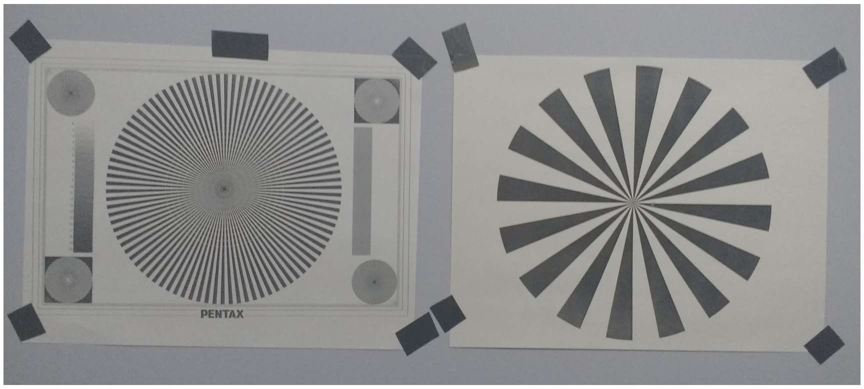


Figure 6.13: Seimens star patterns used to test focused camera images

The Camera intrinsic parameters found from this calibration are as follows:

- **distortion**: [-0.05625918 0.03045371 -0.04082714 0.01728124] +- [0.00820038 0.03371133 0.05421623 0.02943633]
- **projection**: [367.27418734 367.19097289 324.16378311 254.77103486] +- [0.82395318 0.79900595 0.52245299 0.37383656]
- **re-projection error**: [-0.000000, 0.000000] +- [0.153252, 0.141225]

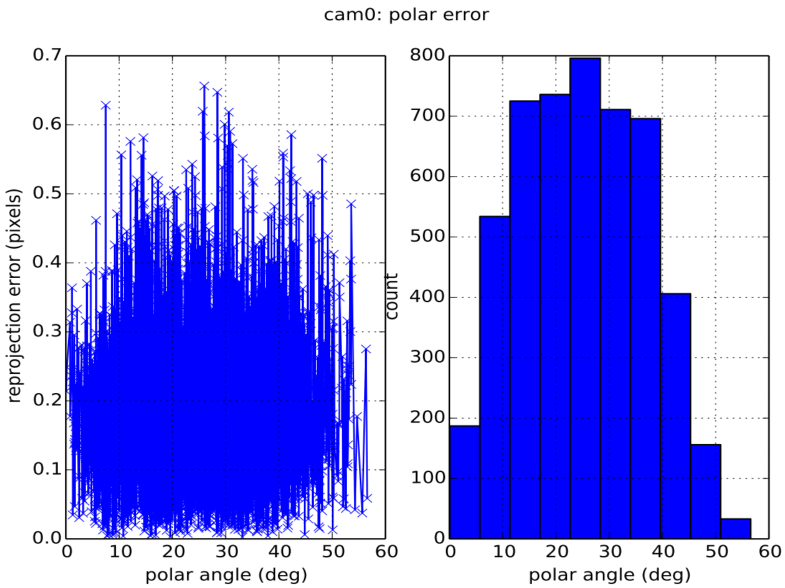


Figure 6.14: Polar error between reprojected pixels after calibration over original observed pixels

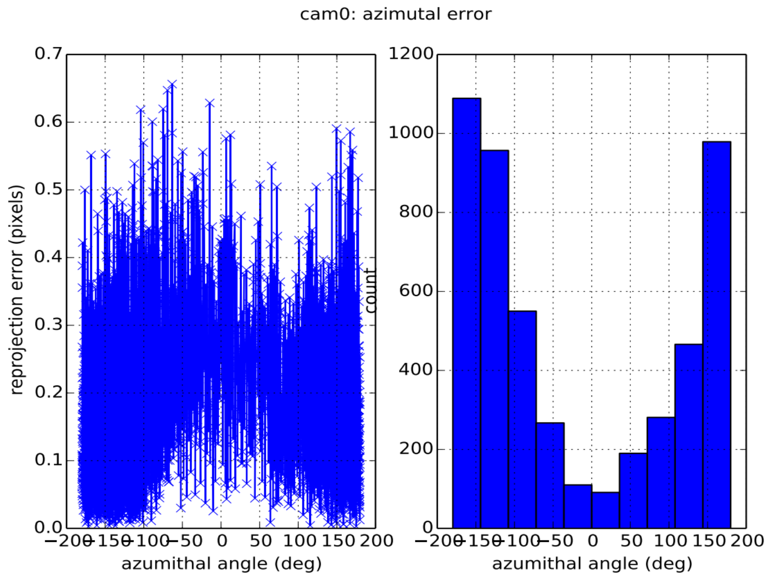


Figure 6.15: Azimuth error of the reprojected pixels after calibration over the original observed pixels.

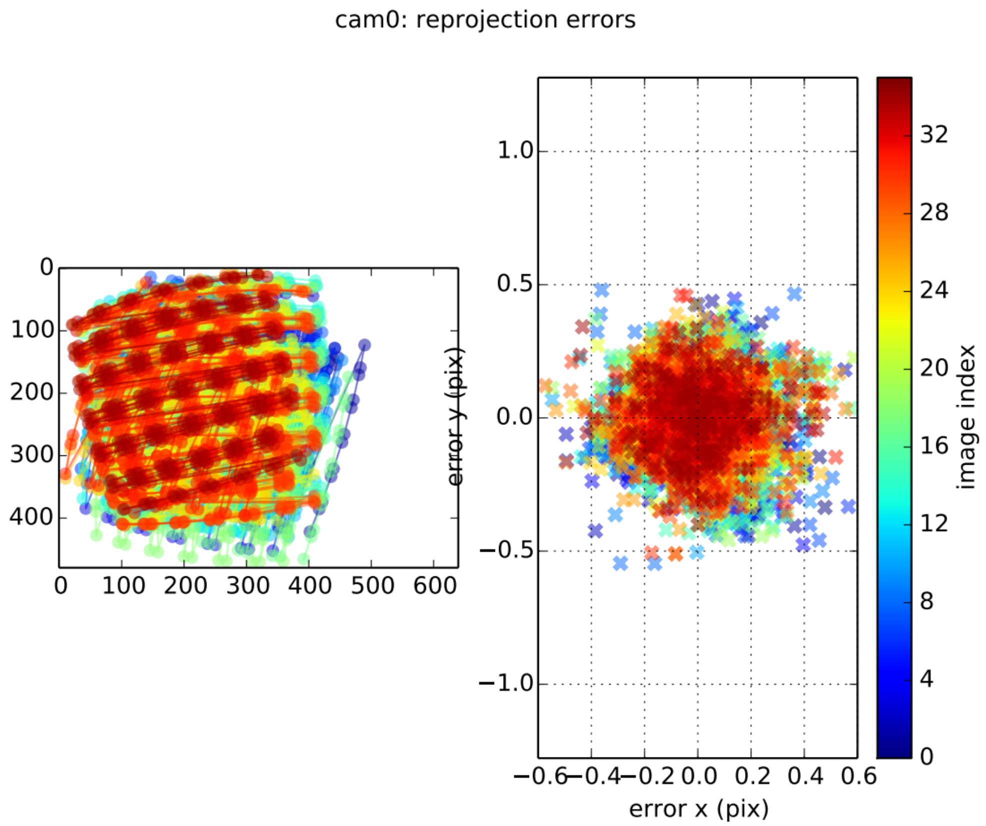


Figure 6.16: Visualization of extracted corners and reprojected pixels errors

6.2.3 Camera-IMU Calibration

By dynamic calibration we mean simultaneous estimation of the motion of the camera-IMU system with respect to static image pixel-pixel correspondences. We use the *Kalibr* toolbox which used high-degree b-spline functions as state variable to estimate continuous motion. After the data is captured, batch-optimization of the data is performed iteratively to calculate the best estimates of time-offsets and the Camera-IMU transformations. For same reasons as static calibration, we list the statistics and results directly.

Accelerometer First the accelerometer overlay-ed on the original data-points are shown in Figure 6.17 in the 3 axes.Figure 6.18 shows the estimation errors and Figure 6.19 shows the estimated bias parameters

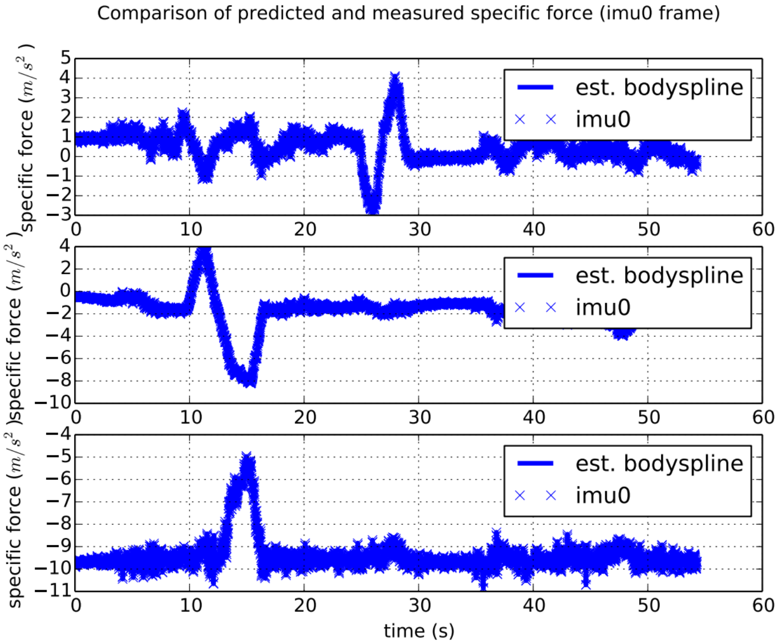


Figure 6.17: Accelerometer curves overlayed over estimated curves

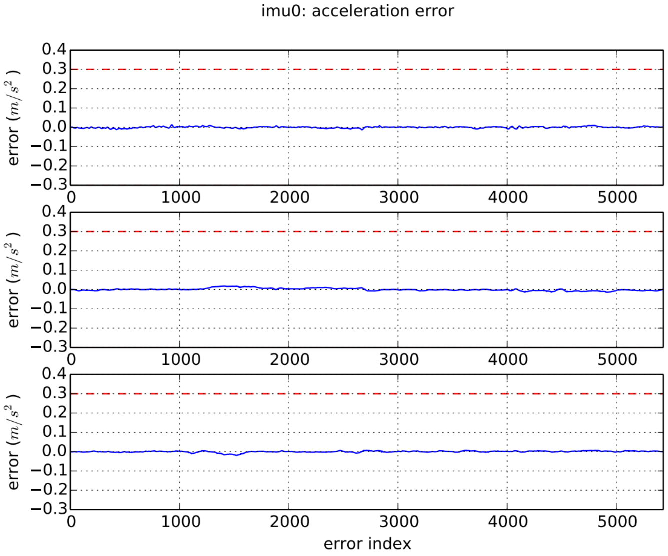


Figure 6.18: Estimated accelerometer bias curve

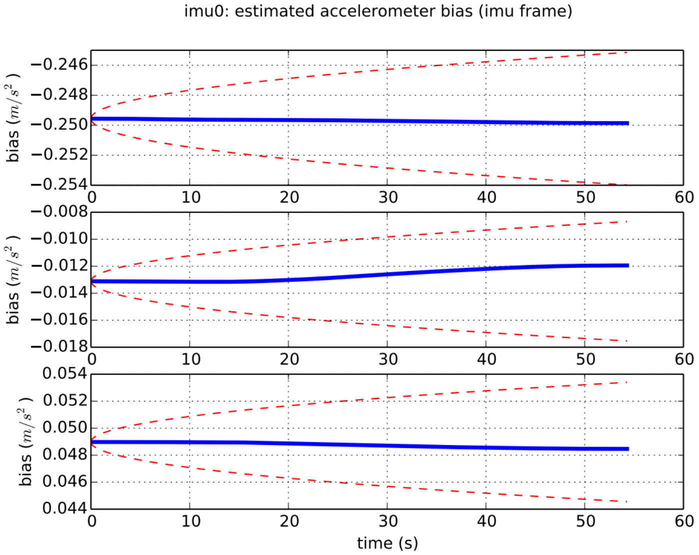


Figure 6.19: Accelerometer bias estimates shown along with upper and lower max-limits

Gyroscope Similar to the analogy above, the estimated values of the gyroscope overlayed on the original data-points are shown in Figure 6.20 in the 3 axes.Figure 6.21 shows the estimation errors and Figure 6.22 shows the estimated bias parameters.

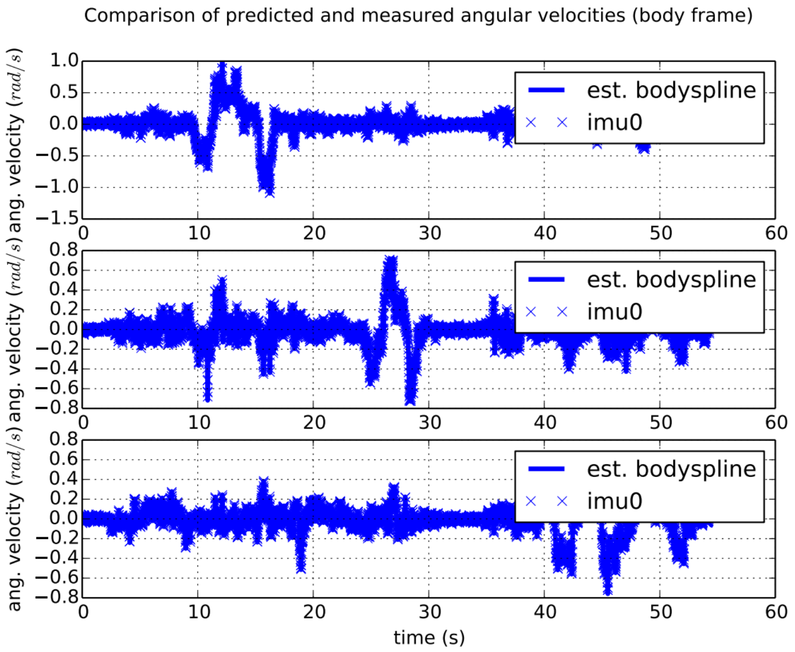


Figure 6.20: Gyroscope curves overlayed over estimated curves

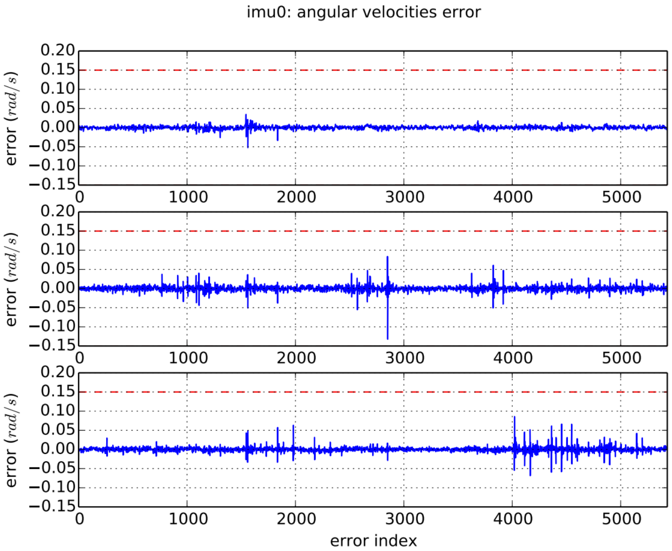


Figure 6.21: Estimated gyroscope bias curve

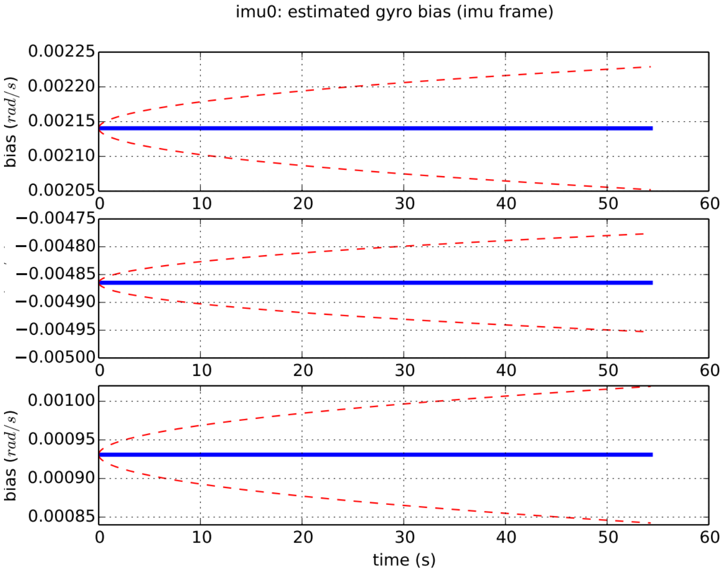


Figure 6.22: Gyroscope bias estimates shown along with upper and lower max-limits

Finally the re-projection error of the dynamic calibration procedure is shown in Figure 6.23. Notice how, with increasing image index(Blue to Red), the re-projection error reduces.

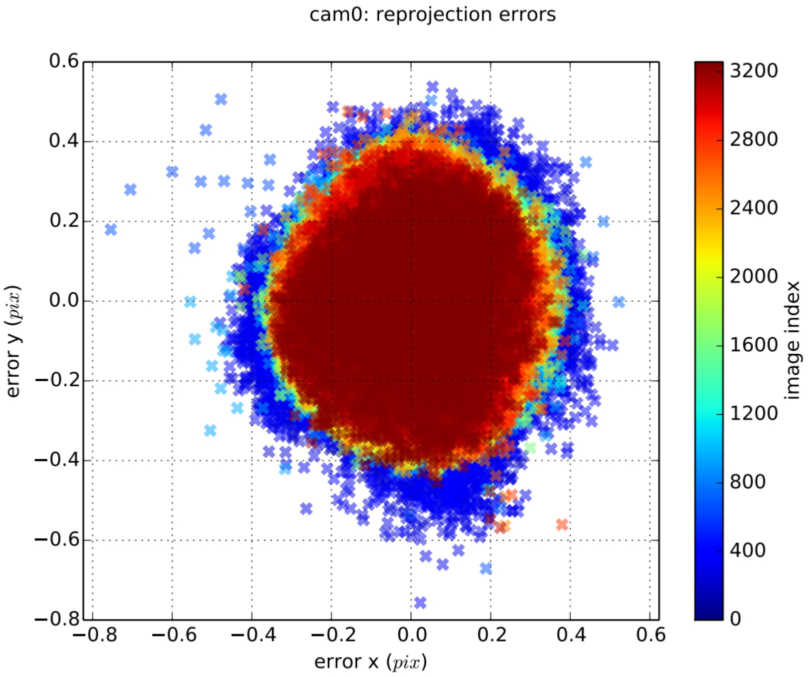


Figure 6.23: Dynamic calibration re-projection error

One thing to be note here is that this is the minimum error we are likely to encounter in our actual estimation algorithm. Also note that this error is much more than only camera re-projection error. This increase in re-projection error was due to the uncertainty associated with the inaccuracy in estimation of camera-IMU transformation Matrix T_{ci}.

Chapter 7

Experimental Results

This chapter outlines the results obtained both indoors and outdoors. Experiments were first conducted indoors where ground-truth was available and later a qualitative evaluation was performed outdoors.

7.1 Indoor Environment

7.1.1 Vicon Room

The accuracy of our algorithm and the tightly coupled approach in the presence of ground-truth data was analyzed The camera-IMU setup was mounted on a trolley with one misaligned wheel which produced unpredictable bumps during movement. To ensure the same conditions for both algorithms, Visual-Inertial Direct (abbreviated as VID) [17] and our method (abbreviated as VIE), were initialized with the same random inverse depth map and the accuracy was analyzed with reference to only one fixed key-frame. Note that the primary motive of our experiment was to observe the initial errors which, in the absence of loop-closure or pose-graph optimization, would persist and accumulate throughout the experiment. The implementation of VID [17] is our own and built on top of [21] instead of [16]. Also, superpixels as implemented in [16], are not included in this comparision.

Table 7.1: Table summarizing accuracy (RMSE) **Translation Errors** of Visual Inertial Direct Method (VID) [17] and the proposed method (VIE), denoted in italics. Also note that the smaller error has been bold-faced for clarity.

Trajectory ID	X(m)		Y(m)		Z(m)	
	VID	*VIE*	VID	*VIE*	VID	*VIE*
1	0.0381	***0.0312***	0.0933	***0.0530***	**0.0569**	*0.0588*
2	0.0273	***0.0237***	**0.0059**	*0.0323*	**0.1154**	*0.1159*
3	**0.0564**	*0.0866*	0.1979	***0.0857***	0.1305	***0.1124***
4	0.0841	***0.0367***	0.0526	***0.0335***	**0.0709**	*0.0988*
5	0.0792	***0.0331***	0.1333	***0.0670***	0.0554	***0.0512***
6	0.0939	***0.0411***	0.0739	***0.0376***	0.1203	***0.0756***

Table 7.2: Table summarizing accuracy (RMSE) **Rotation Errors** of Visual Inertial Direct Method (VID) [17] and the proposed method (VIE), denoted in italics. Also note that the smaller error has been bold-faced for clarity.

Trajectory ID	Yaw(rad)		Pitch(rad)		Roll(rad)	
	VID	*VIE*	VID	*VIE*	VID	*VIE*
1	0.0213	***0.0038***	0.0635	***0.0136***	0.0426	***0.0348***
2	0.2805	***0.0020***	0.0456	***0.0181***	0.0284	***0.0025***
3	0.3528	***0.0131***	0.0832	***0.0253***	**0.0628**	*0.1066*
4	0.7987	***0.0013***	0.0567	***0.0225***	0.0416	***0.0200***
5	0.0138	***0.0112***	0.0295	***0.0149***	**0.0243**	*0.0667*
6	0.3134	***0.0037***	**0.0269**	*0.0321*	**0.0175**	*0.0265*

Since our long term objective was to develop this system for land-vehicles, the movement was limited to the two-dimensional plane only. Moreover, complex 3D trajectories are usually observed when mounted on drones or simply hand-held mapping systems, where the noise profile due to wind-gusts or hand tremor is much different than what is observed in land-vehicles.

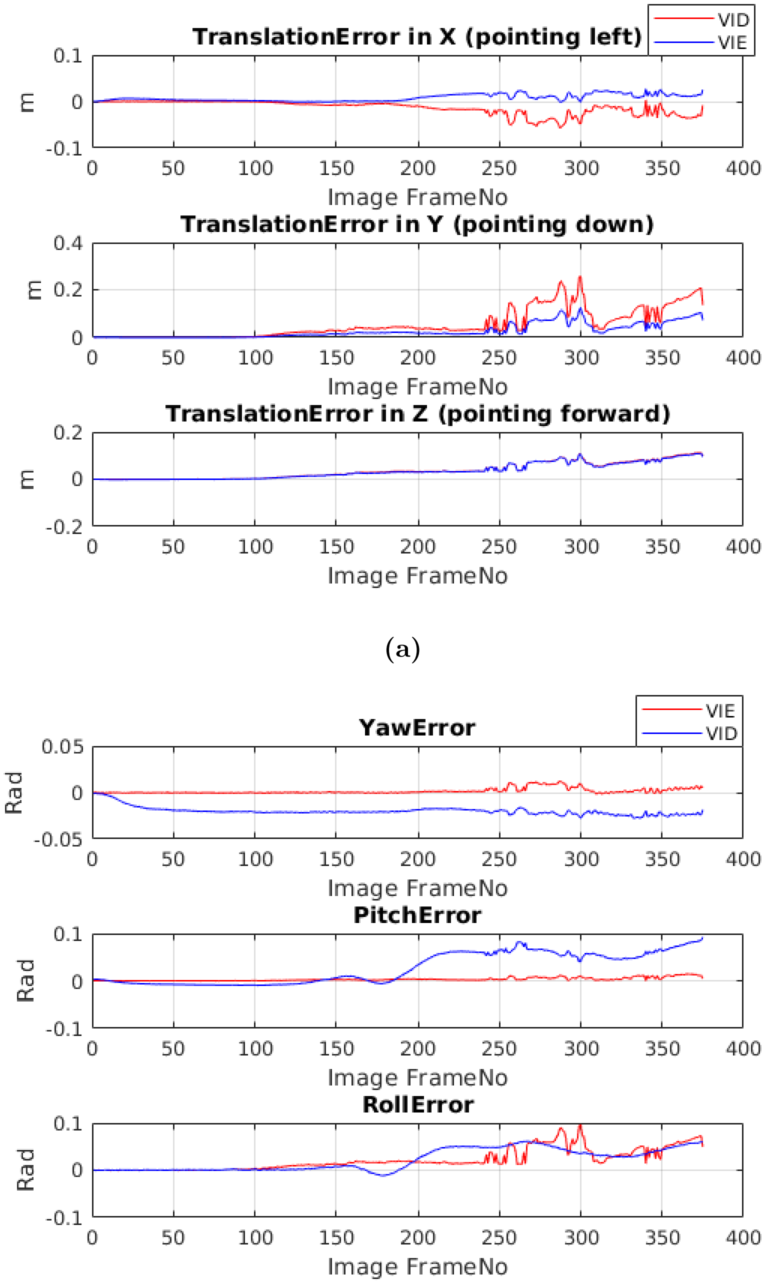
The platform was moved several times in presence of ground-truth in different directions and the results (RMSE errors) are summarized in Table 7.1 and 7.2.

On closer inspection of the results in Table 7.1 and 7.2, one can observe that overall, better accuracy is achieved using our method than the state-of-the-art (∼26% improvement in translation and $\sim 55\%$ improvement in rotation), in presence of sudden spikes in accelerations due to bumps.

By looking at the raw IMU measurements Figures 7.2, 7.4, 7.6, 7.8, 7.10, 7.12, one can easily spot the the time instants where the trolley cart experienced sudden bumps (areas of high oscillations). One can observe from raw accelerometer readings in Figures 7.2a , 7.4a, 7.6a, 7.8a, 7.10a and 7.12a, that the magnitude of the noise is dominant in the downward facing 'Z' direction, although the lateral 'X' and 'Y' direction measurements suffer as well. By looking at the raw gyroscope readings in Figure 7.2b 7.4b, 7.6b, 7.8b, 7.10b and 7.12b, one can deduce that the noise due to bumps affects angular measurements as well. Yaw (g_z) remains relatively noise free while pitch and roll are impacted greatly as a result of bumps.

From the plots (Figures 7.1a and 7.1b), it is evident that as the cart progressively experienced more bumps, the tightly-coupled system's (VID) accuracy in pose estimation degraded. At this point, qualitative correlation can be drawn between the raw IMU readings in Figure 7.2 and the effect it had on accuracy of the two algorithms in Figure 7.1. By decoupling the IMU from the joint optimization step, the noise could be reduced in Y (pointing down) and X (pointing sideways from direction of motion). As the movement of the cart was perpendicular to the observing surface (wall directly in front), our system was unable to eliminate the noise component in the Z (direction of motion). It can also be seen that noise due to bumps not only affected the translation but rotation as well (as the optimization was jointly performed over $\mathbf{SE(3)}$). Similar inferences can be drawn from plots for other trajectories.

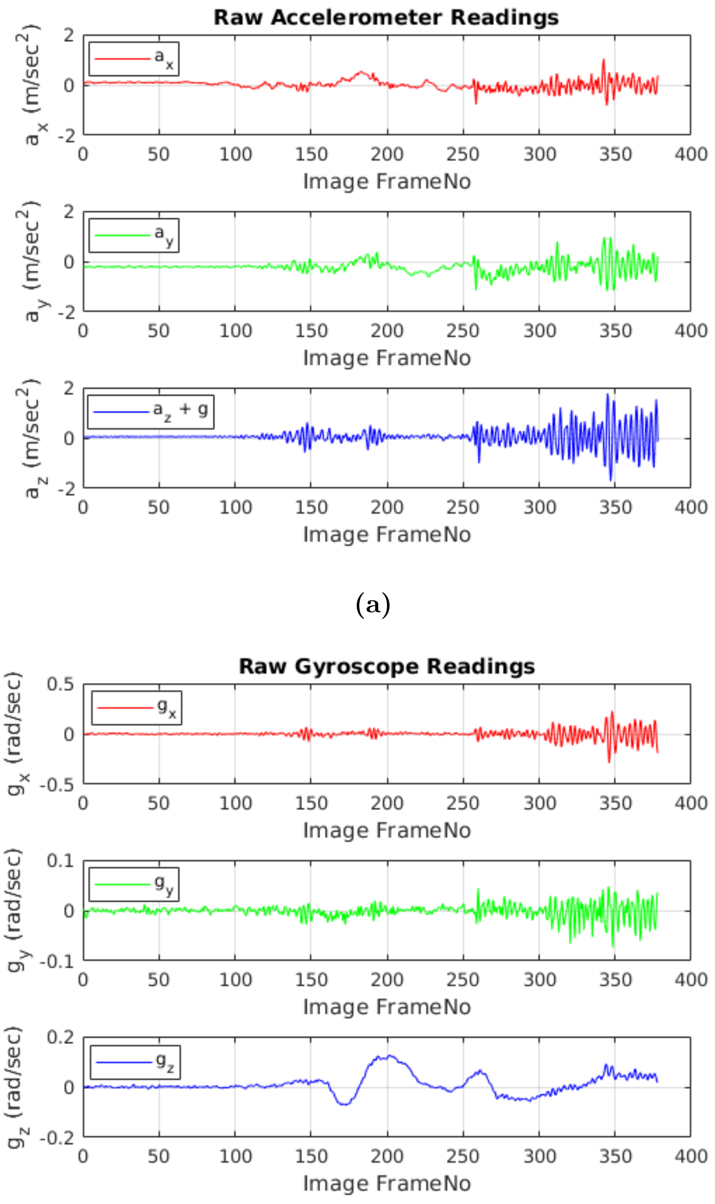
Further, it can be observed that trajectory 2 in Table 7.1 shows better translation accuracy for the competing method. On comparision against IMU acceleration profile for other trajectories, one can observe a lower magnitude of spike in Figure 7.4, suggesting that in the absence of large spikes, the performance of the VID technique is better, as expected, due to tight-coupling. Moreover, the subtle vibrations seen in the plots (Figures 7.1a and 7.1b) are a direct result of the camera capturing bumps at its frame-rate, which shows up in both the techniques. However, vibrations induced in-between two camera frames could be corrected using the proposed method in contrast to the VID technique.



(a)

(b)

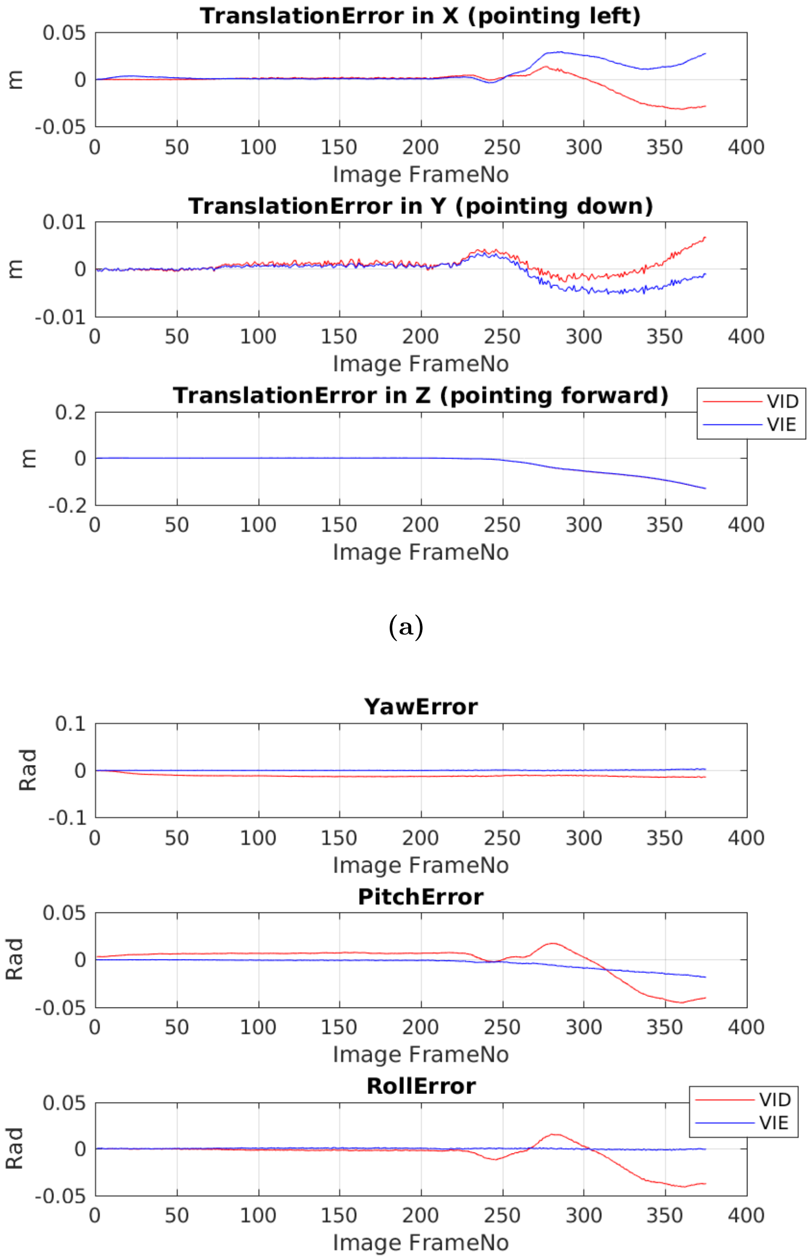
Figure 7.1: Trajectory 1:(a) Translation errors(m) versus image frame number (b) Angular errors(rad) versus Image Frame number. Note: The coordinate frame expressed here is Camera centric. Z-forward, Y-down and X-right.



(a)

(b)

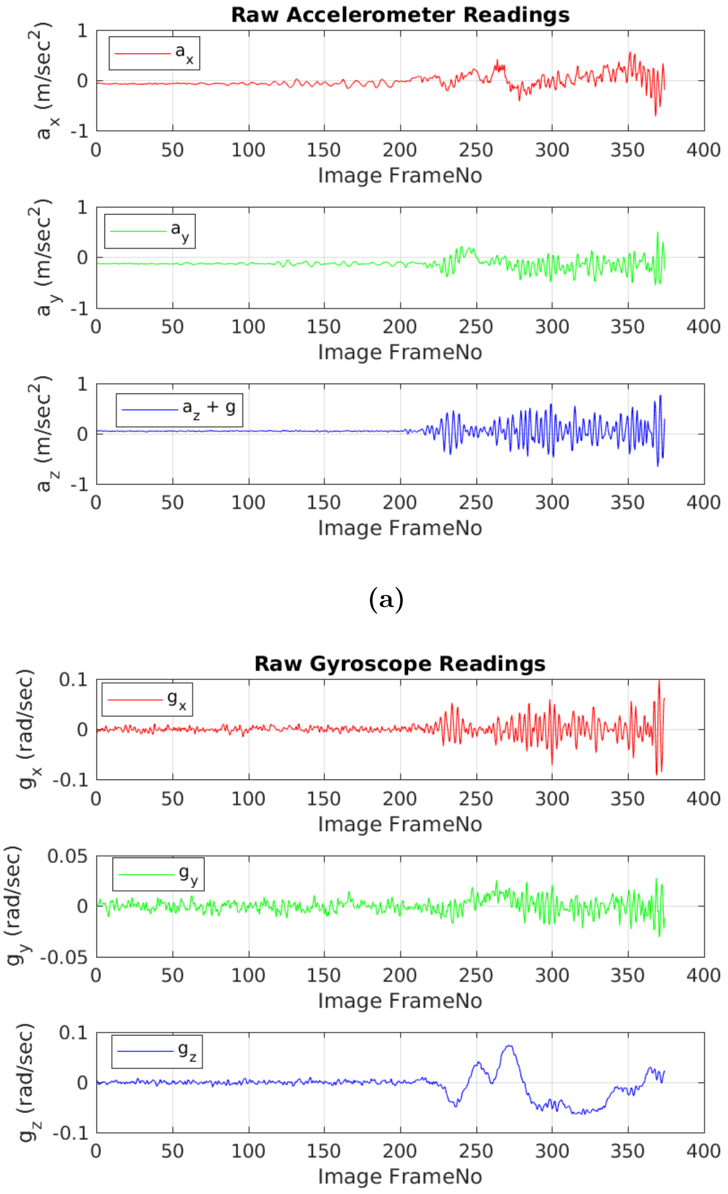
Figure 7.2: Trajectory 1:(a) Raw accelerometer reading and (b) Raw gyroscope reading versus Image frame number. Note, that even though IMU sampling rate(100Hz) is twice that of the camera(50Hz), the readings are plotted with respect to Image Frame No. for easy comparison. Also note that the coordinate system in IMU centric. X-forward, Y-Right, Z-Down



(a)

(b)

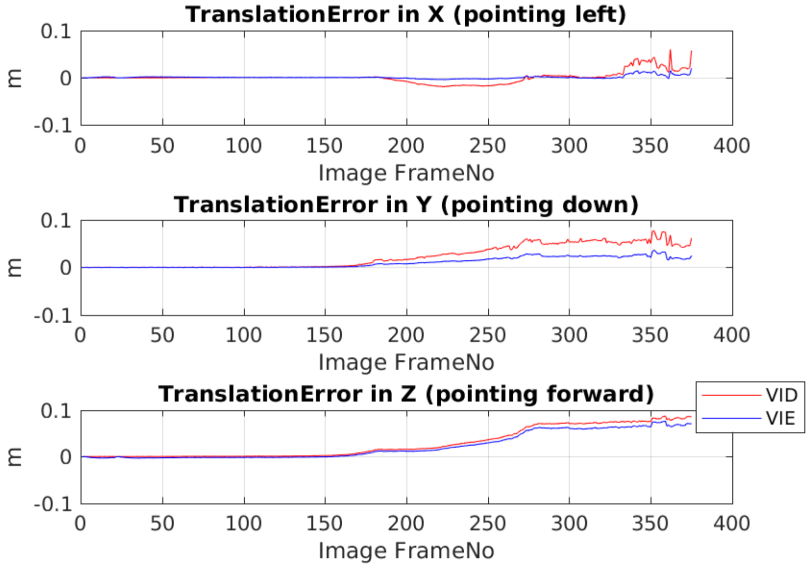
Figure 7.3: Trajectory 2:(a)Translation errors(m) versus image frame number (b) Angular errors(rad) versus image frame number. Note: The coordinate frame expressed here is camera centric. Z-forward, Y-down and X-right.



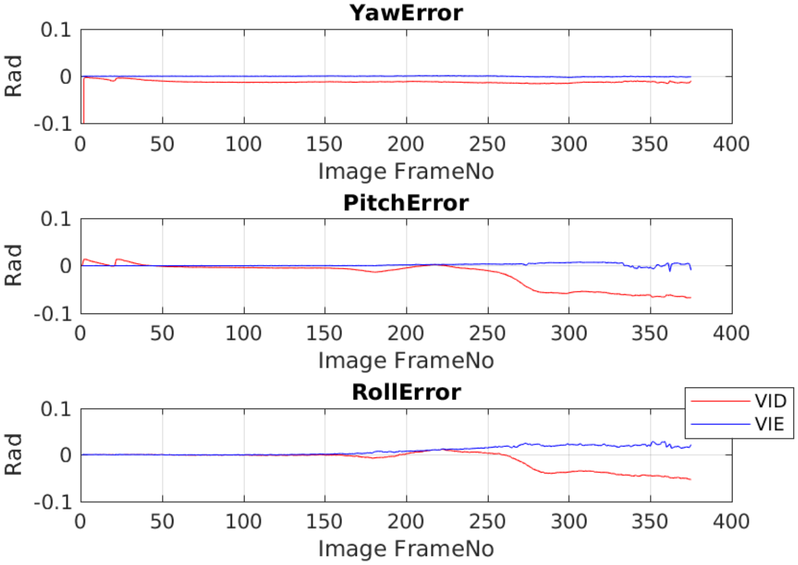
(a)

(b)

Figure 7.4: Trajectory 2:(a) Raw accelerometer reading and (b) Raw gyroscope reading versus Image frame number. Note, that even though IMU sampling rate(100Hz) is twice that of the camera(50Hz), the readings are plotted with respect to Image Frame No. for easy comparison. Also note that the coordinate system in IMU centric. X-forward, Y-Right, Z-Down

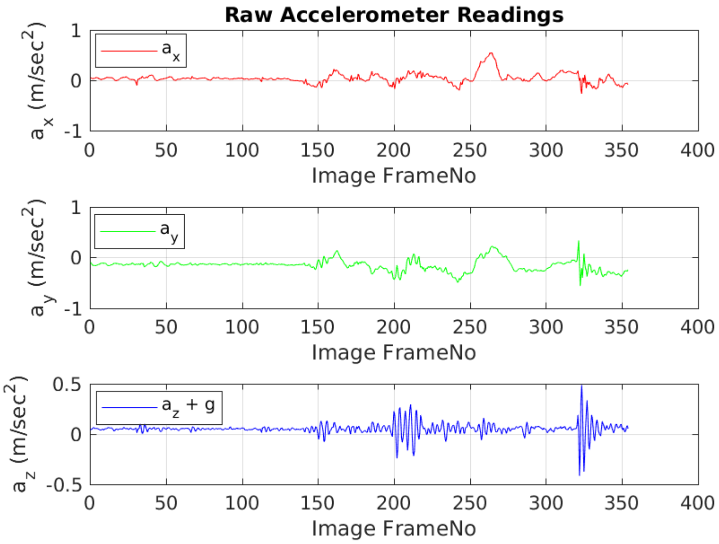


(a)

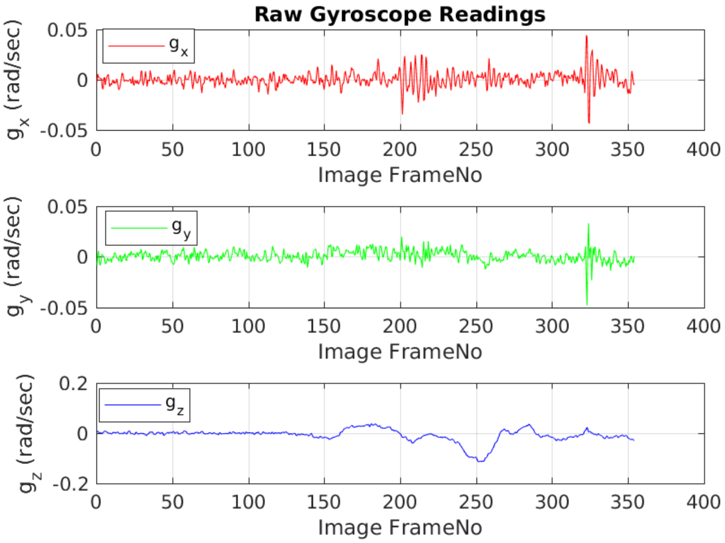


(b)

Figure 7.5: Trajectory 3:(a) Translation errors(m) versus image frame number (b) Angular errors(rad) versus image frame number. Note: The coordinate frame expressed here is camera centric. Z-forward, Y-down and X-right.

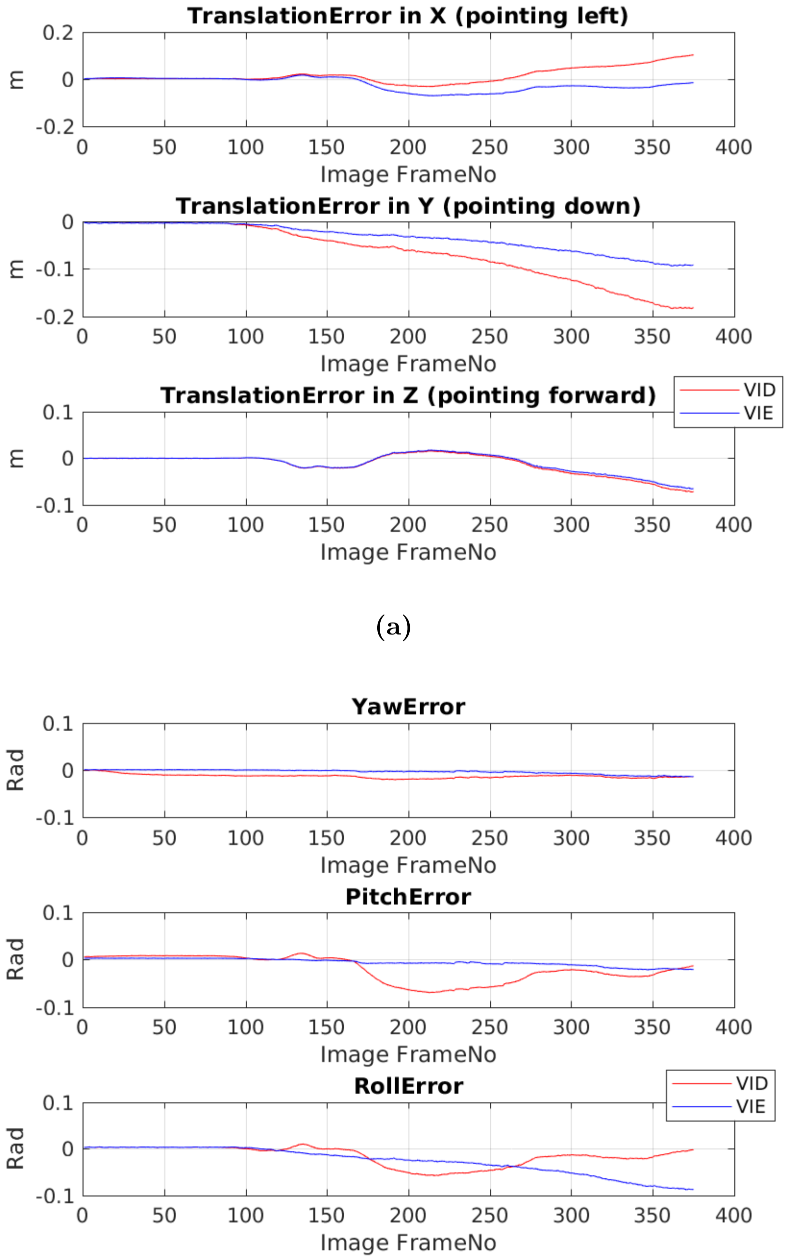


(a)



(b)

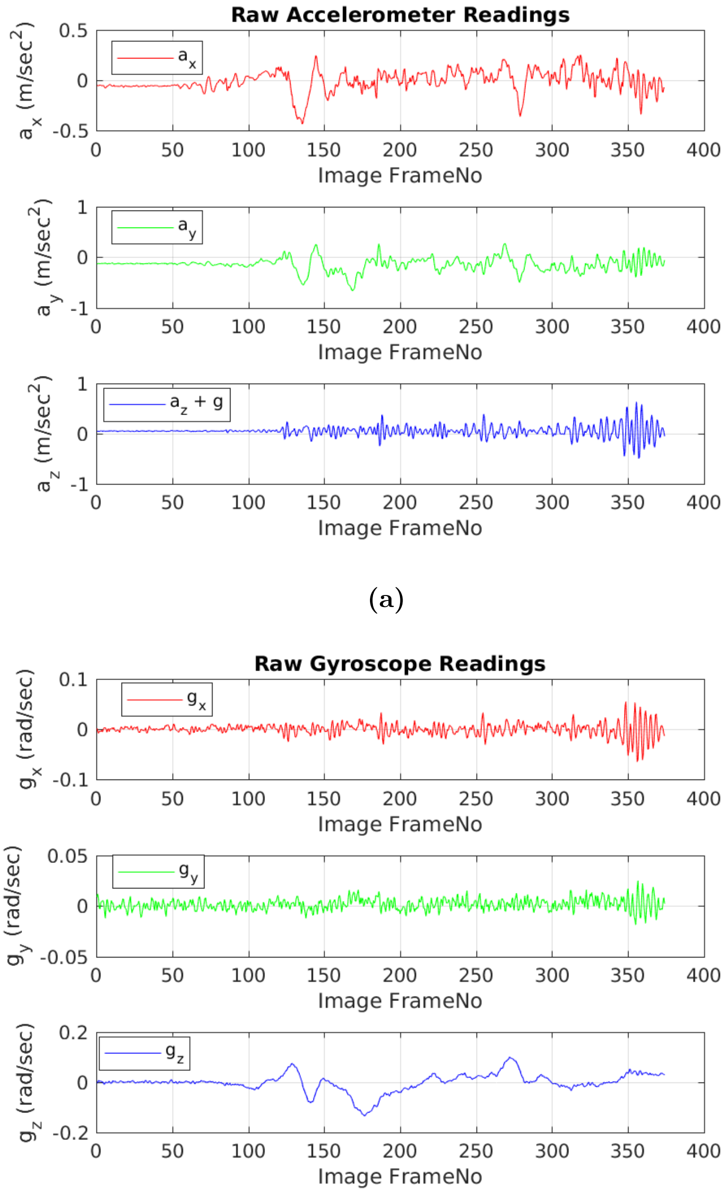
Figure 7.6: Trajectory 3:(a) Raw accelerometer reading and (b) Raw gyroscope reading versus Image frame number. Note, that even though IMU sampling rate(100Hz) is twice that of the camera(50Hz), the readings are plotted with respect to Image Frame No. for easy comparison. Also note that the coordinate system in IMU centric. X-forward, Y-Right, Z-Down



(a)

(b)

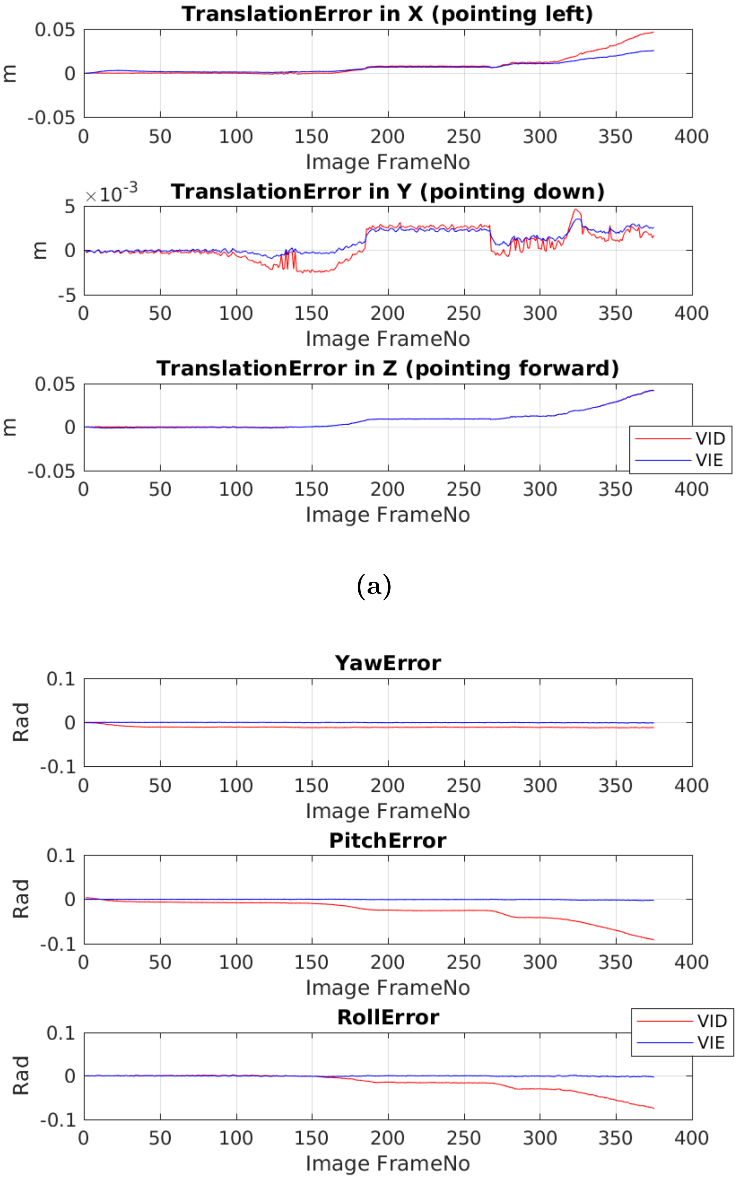
Figure 7.7: Trajectory 4:(a) Translation errors(m) versus image frame number (b) Angular errors(rad) versus image frame number. Note: The coordinate frame expressed here is camera centric. Z-forward, Y-down and X-right.



(a)

(b)

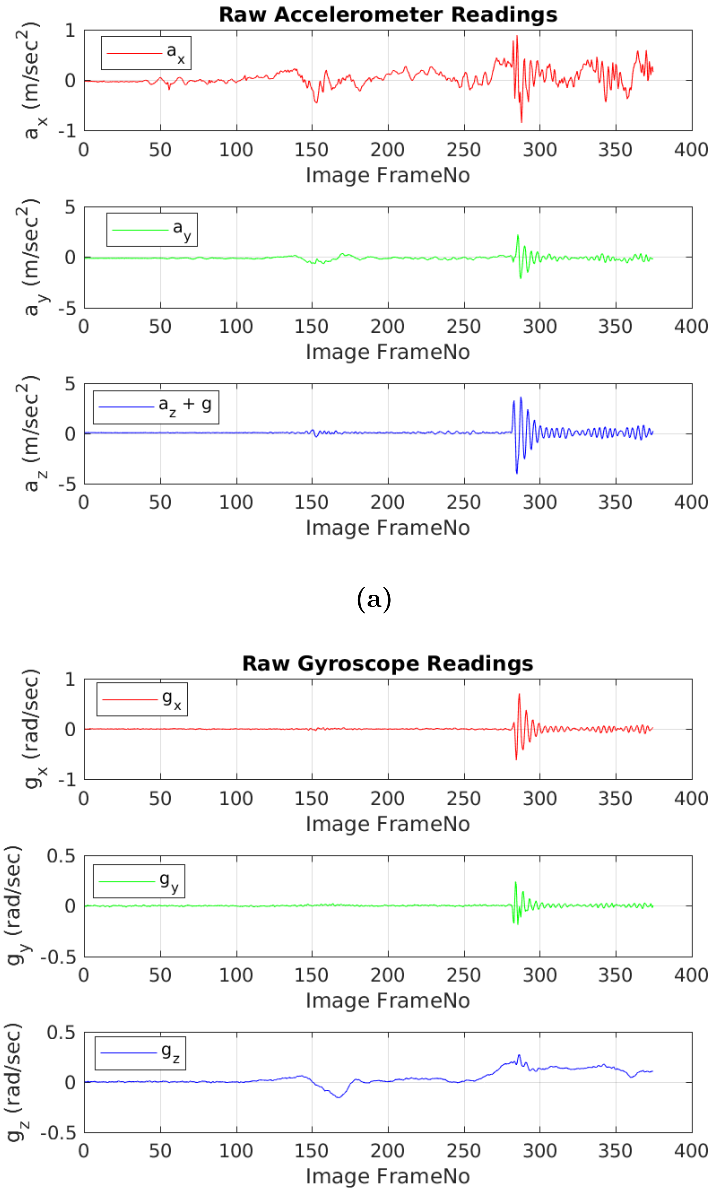
Figure 7.8: Trajectory 4:(a) Raw accelerometer reading and (b) Raw gyroscope reading versus Image frame number. Note, that even though IMU sampling rate(100Hz) is twice that of the camera(50Hz), the readings are plotted with respect to Image Frame No. for easy comparison. Also note that the coordinate system in IMU centric. X-forward, Y-Right, Z-Down



(a)

(b)

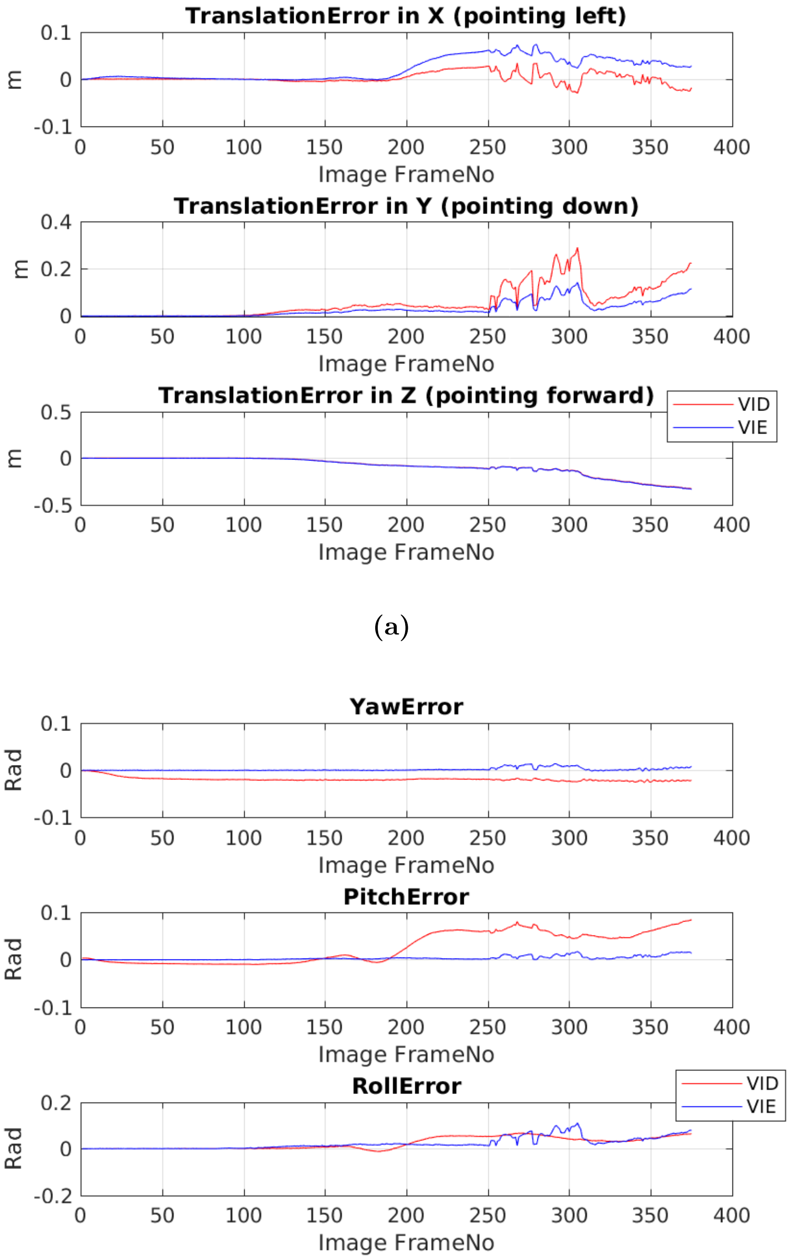
Figure 7.9: Trajectory 5:(a) Translation errors(m) versus image frame number (b) Angular errors(rad) versus image frame number. Note: The coordinate frame expressed here is camera centric. Z-forward, Y-down and X-right.



(a)

(b)

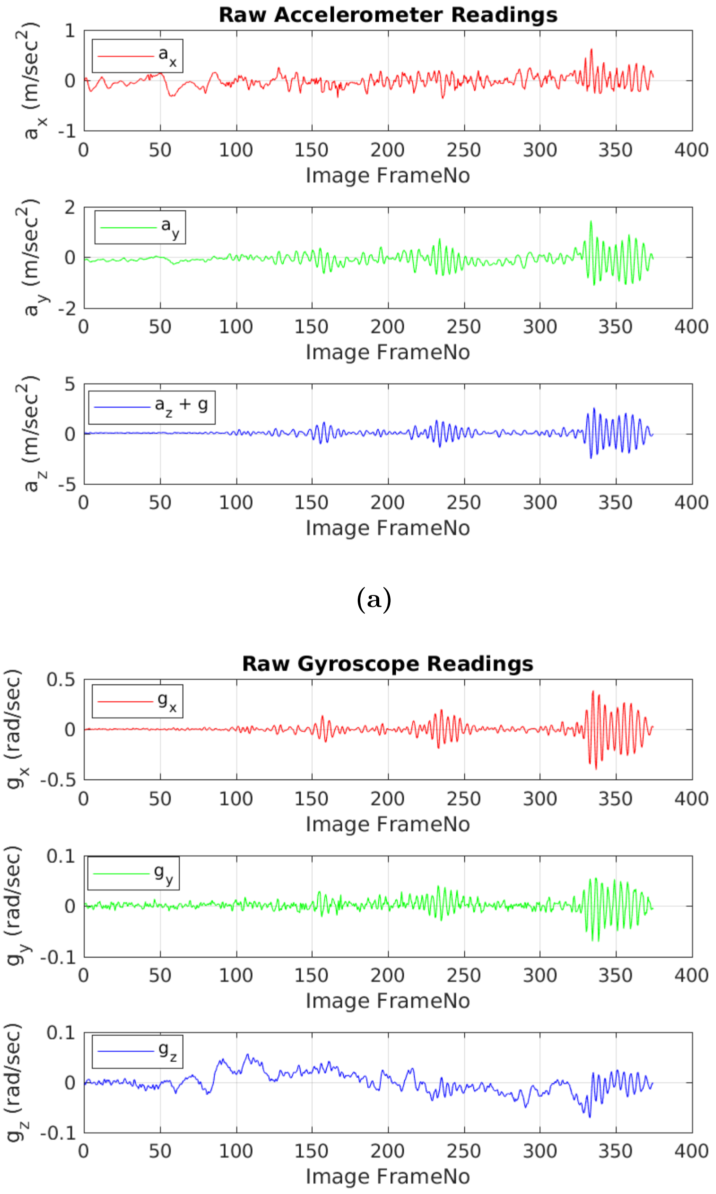
Figure 7.10: Trajectory 5:(a) Raw accelerometer reading and (b) Raw gyroscope reading versus Image frame number. Note, that even though IMU sampling rate(100Hz) is twice that of the camera(50Hz), the readings are plotted with respect to Image Frame No. for easy comparison. Also note that the coordinate system in IMU centric. X-forward, Y-Right, Z-Down



(a)

(b)

Figure 7.11: Trajectory 6:(a) Translation errors(m) versus image frame number (b) Angular errors(rad) versus image frame number. Note: The coordinate frame expressed here is camera centric. Z-forward, Y-down and X-right.



(a)

(b)

Figure 7.12: Trajectory 6:(a) Raw accelerometer reading and (b) Raw gyroscope reading versus Image frame number. Note, that even though IMU sampling rate(100Hz) is twice that of the camera(50Hz), the readings are plotted with respect to Image Frame No. for easy comparison. Also note that the coordinate system in IMU centric. X-forward, Y-Right, Z-Down

7.1.2 Corridor

After validating our method in the indoor setting with ground-truth, the same trolley-camera-IMU system was used to map a larger in indoor area. The map generated is shown in Figure 7.13. The quality of the map, even in the presence of bumpy motion is a demonstration of the pose-estimation accuracy of our approach.

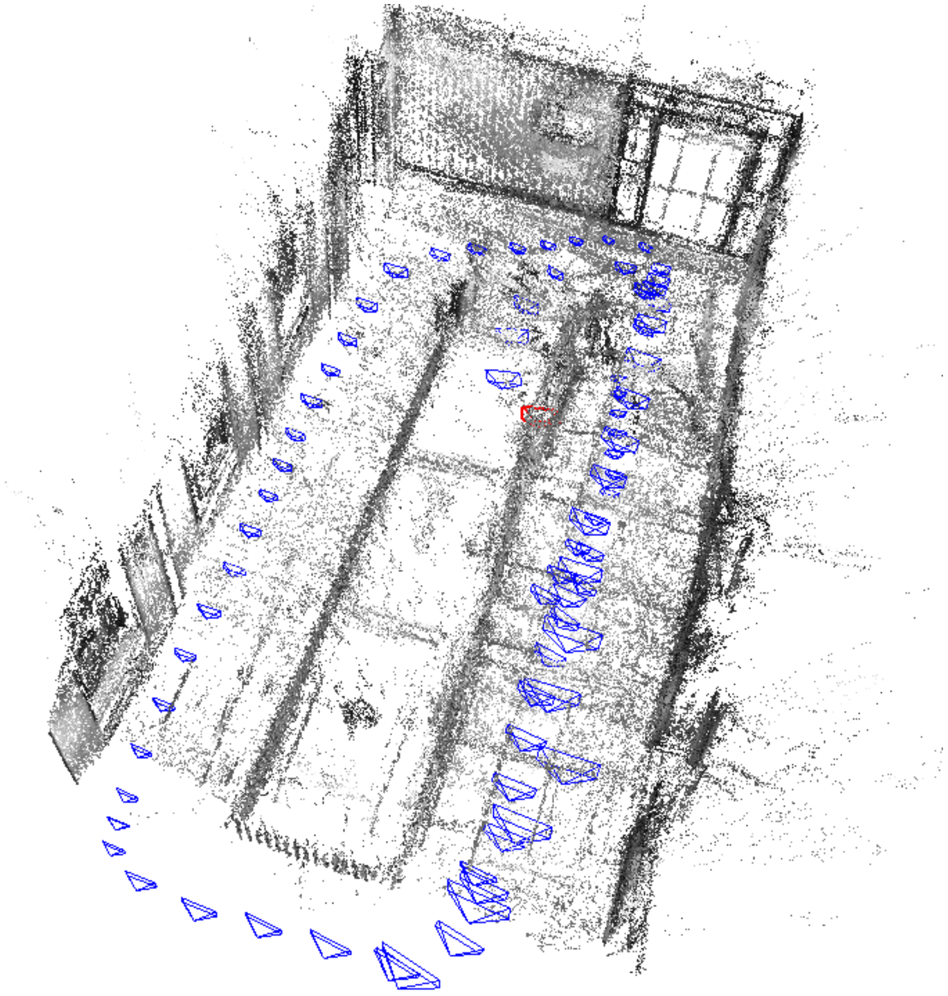
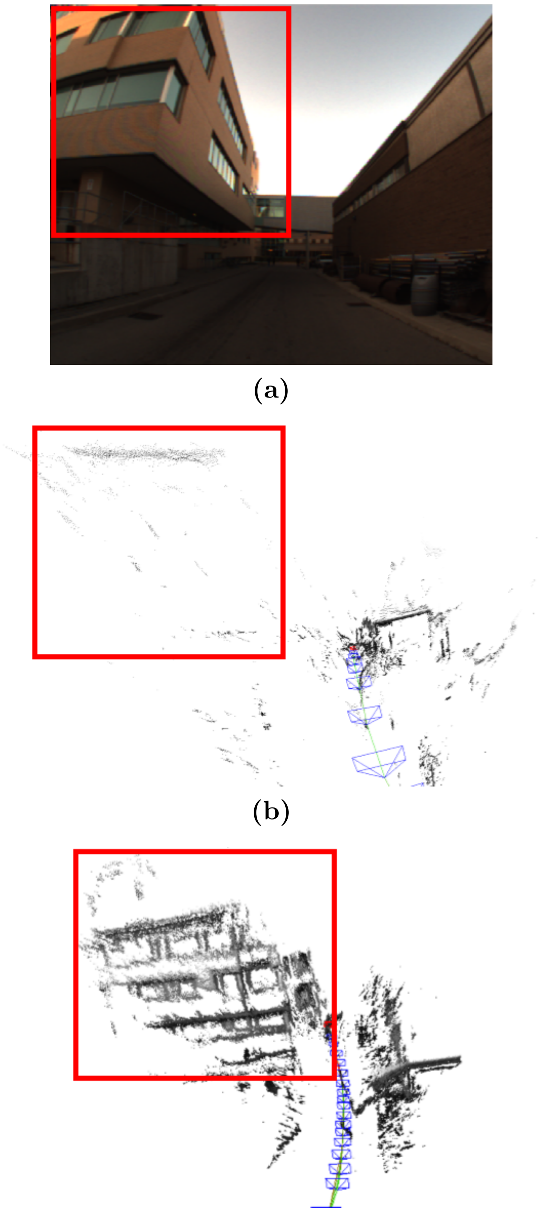


Figure 7.13: A semi-dense map build of an indoor corridor.

7.2 Outdoor Environment

The camera-IMU system was mounted on the ARGO 6x6 off-road land-vehicle. The set-up was subject to high noise due to vibration of the vehicle chassis, due to sudden acceleration and braking and during general motion on the road terrain.

Since there was no way to estimate the ground truth pose accurately, the results shown here are only qualitative. A portion of the environment (a building) in the RGB image is highlighted as seen in Figure 7.14a, the same region as reconstructed using tightly coupled approach in Figure 7.14b and using our method in Figure 7.14c. A significant degradation in map quality due to tight coupling in presence of high inertial noise can be noticed. As mapping is done in a SLAM framework, the error in pose prediction affects the quality of map that is built consequently. Since our approach is resilient to high inertial noise (as shown in an indoor settings , refer to Section 7.1.1), the quality of the map built using our technique was superior.



(a)

(b)

(c)

Figure 7.14: Qualitative results of outdoor experiment. A portion of the 3D structure is highlighted in RED in all three figures for comparison. (a) shows the sample RGB image seen by the camera. (b) shows reconstruction quality for tightly-coupled system. (c) shows reconstruction quality for the proposed method. Notice the improvement in map quality due to increased accuracy of pose estimation.

Chapter 8

Conclusion and Future Work

In this work, a semi-tightly coupled direct visual-inertial fusion scheme was presented to handle sudden, unintended bumps encountered when the camera-IMU system was mounted on a land-vehicle. The multitude of visual correspondences provided enough constraints to correct large inter-frame IMU drifts. Further, by accounting for inverse-depth variances in the optimization framework, we could include information from all valid pixels in our inertial-epipolar optimization, making our fusion method a direct-approach. Although, an IMU has traditionally been used to speed up the prediction in a tightly-coupled framework, through experiments it was shown that a wrong prior at the start made the joint optimization objective converge to a local minima. Hence, it was reasonable to isolate the IMU measurements and correct it later by imposing epipolar constraints.

Experiments were first conducted indoors, in the presence of ground-truth and compared with the current tightly-coupled state-of-the-art visual-inertial method to demonstrate increase in accuracy of pose-prediction. To simulate unintended bumps, a makeshift trolley with one mis-aligned wheel was used. The inability to control the duration or timing of the bumps not only made each experiment unique but also mimic outdoor scenarios. The experiments were repeated for six trajectories in an indoor environment, in order to confirm the validity of this approach. On close inspection of the plots, not only positional but also rotational accuracy improvement can be noticed.

Experiments conducted outside were only quantitative as there was no way to measure ground-truth. The camera-IMU setup suffered from high noise both because of vibrations on the chassis and uneven road terrain. It was demonstrated that the proposed method built a reasonable map of the environment when the competing method quickly diverged.

However, since the proposed approach uses two optimization objectives instead of one, it required a minor computational overhead ($\sim$ 10 iterations, 12±5ms), while still achieving real-time speed. A trade-off in speed was the price paid to combat noise due to bumps. In the future, the proposed technique can be equipped with loop closure and re-localization to further improve the accuracy. Further, as visual point-cloud contains color/brightness information, semantic segmentation can be done for object labeling, collision free-path generation, etc.

The proposed technique is best suited for off-road land vehicles which are prone to sudden bumps and change of terrain. However, in cases where computational resource is limited and the noise due to motion can be appropriately modeled, the tightly-coupled approach may be used. The author believes that this work will find useful application for state-estimation of land vehicles, especially in off-road environments.

Bibliography

[1] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*, pages 2320–2327, 11 2011.

[2] Sameer Agarwal, Keir Mierle, and Others. Ceres solver a large scale non-linear optimization library.

[3] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. *IEEE Robotics Automation Magazine*, 13(3):108–117, Sept 2006.

[4] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, Feb 2004.

[5] S. Y. Bao, M. Bagra, Y. Chao, and S. Savarese. Semantic structure from motion with points, regions, and objects. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2703–2710, June 2012.

[6] Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart. Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072, 2017.

[7] Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart. Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072, 2017.

[8] H. Bradler, M. Ochs, N. Fanani, and R. Mester. Joint epipolar tracking (jet): Simultaneous optimization of epipolar geometry and feature correspondences. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 445–453, March 2017.

[9] Michael Burri, Helen Oleynikova, Markus W. Achtelik, and Roland Siegwart. Real-time visual-inertial mapping, re-localization and planning onboard mavs in unknown environments.

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

[11] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert. Initialization techniques for 3d slam: A survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4597–4604, May 2015.

[12] D. Caruso, J. Engel, and D. Cremers. Large-scale direct slam for omnidirectional cameras. In *International Conference on Intelligent Robots and Systems (IROS)*, September 2015.

[13] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular slam. *IEEE Transactions on Robotics*, 24(5):932–945, Oct 2008.

[14] Javier Civera, Andrew J. Davison, and J. M. Montiel. Dimensionless monocular slam. In *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis, Part II*, IbPRIA '07, pages 412–419, Berlin, Heidelberg, 2007. Springer-Verlag.

[15] A. Concha and J. Civera. Using superpixels in monocular slam. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 365–372, May 2014.

[16] A. Concha and J. Civera. Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5686–5693, Sept 2015.

[17] A. Concha, G. Loianno, V. Kumar, and J. Civera. Visual-inertial direct slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1331–1338, May 2016.

[18] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.

[19] US Department of Transportaion. How much time do americans spend behind the wheel?, Dec 2017.

[20] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, June 2006.

[21] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*, September 2014.

[22] J. Engel, J. Stueckler, and D. Cremers. Large-scale direct slam with stereo cameras. In *International Conference on Intelligent Robots and Systems (IROS)*, September 2015.

[23] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[24] J. Engel, J. Sturm, and D. Cremers. Scale-aware navigation of a low-cost quadrocopter with a monocular camera. *Robotics and Autonomous Systems (RAS)*, 62(11):1646–1656, 2014.

[25] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, Dec 2001.

[26] Association for safe international Road Travel. Annual global road crash statistics.

[27] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, Feb 2017.

[28] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, April 2017.

[29] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, Jan 2015.

[30] P. Furgale, T. D. Barfoot, and G. Sibley. Continuous-time batch estimation using temporal basis functions. In *2012 IEEE International Conference on Robotics and Automation*, pages 2088–2095, May 2012.

[31] P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286, Nov 2013.

[32] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, June 2011.

[33] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[34] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[35] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*, 2010.

[36] W. N. Greene, K. Ok, P. Lommel, and N. Roy. Multi-level mapping: Real-time dense monocular slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 833–840, May 2016.

[37] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, Apr 2013.

[38] Peter J. Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35(1):73–101, 03 1964.

[39] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik G. Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. *CoRR*, abs/1712.00080, 2017.

[40] Eagle S. Jones and Stefano Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011.

[41] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vslam algorithm for robust localization and mapping. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 24–29, April 2005.

[42] Jonathan Kelly and Gaurav S Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1):56–79, 2011.

[43] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[44] Frank R Kschischang, Brendan J Frey, and Hans Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[45] R. Kummerle, D. Hahnel, D. Dolgov, S. Thrun, and W. Burgard. Autonomous driving in a multi-level parking structure. In *2009 IEEE International Conference on Robotics and Automation*, pages 3395–3400, May 2009.

[46] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization.

[47] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visualinertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[48] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, June 2011.

[49] Mingyang Li and Anastasios I. Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[50] T. Liu and S. Shen. High altitude monocular visual-inertial state estimation: Initialization and sensor fusion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4544–4551, May 2017.

[51] Y. Liu, R. Xiong, Y. Wang, H. Huang, X. Xie, X. Liu, and G. Zhang. Stereo visual-inertial odometry with multiple kalman filters ensemble. *IEEE Transactions on Industrial Electronics*, 63(10):6205–6216, Oct 2016.

[52] F. Maurelli, D. Droeschel, T. Wisspeintner, S. May, and H. Surmann. A 3d laser scanner system for autonomous vehicle navigation. In *2009 International Conference on Advanced Robotics*, pages 1–6, June 2009.

[53] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4628–4635, May 2017.

[54] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572, April 2007.

[55] R. Mur-Artal and J. D. Tards. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, April 2017.

[56] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[57] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[58] J. Mustaniemi, J. Kannala, S. Srkk, J. Matas, and J. Heikkil. Inertial-based scale estimation for structure from motion on mobile devices. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4394–4401, Sept 2017.

[59] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Oct 2011.

[60] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar. Visual odometry system using multiple stereo cameras and inertial measurement unit. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[61] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3491–3500, July 2017.

[62] T. Qin and S. Shen. Robust initialization of monocular visual-inertial estimation on aerial robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4225–4232, Sept 2017.

[63] J. A. Smith, T. L. Lin, and K. L. Ranson. The lambertian assumption and landsat data. *Photogrammetric Engineering and Remote Sensing*, 46(9):1183–1189, 1980.

[64] Shiyu Song and Manmohan Chandraker. Robust scale estimation in real-time monocular sfm for autonomous driving. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[65] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Scale drift-aware large scale monocular slam. In *Robotics: Science and Systems*, 2010.

[66] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, Jun 2017.

[67] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692.

[68] Christopher Urmson, Joshua Anhalt, J. Andrew (Drew) Bagnell, Christopher R. Baker, Robert E. Bittner, John M. Dolan, David Duggins, David Ferguson, Tugrul Galatali, Hartmut Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas Howard, Alonzo Kelly, David Kohanbash, Maxim Likhachev, Nick Miller, Kevin Peterson, Raj Rajkumar, Paul Rybski, Bryan Salesky, Sebastian Scherer, Young-Woo Seo, Reid Simmons, Sanjiv Singh, Jarrod M. Snider, Anthony (Tony) Stentz, William (Red) L. Whittaker, and Jason Ziglar. Tartan racing: A multi-modal approach to the darpa urban challenge. Technical report, Carnegie Mellon University, Pittsburgh, PA, April 2007.

[69] V. Usenko, J. Engel, J. Stckler, and D. Cremers. Direct visual-inertial odometry with stereo cameras. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1885–1892, May 2016.

[70] V. Usenko, L. von Stumberg, A. Pangercic, and D. Cremers. Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer. In *International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, Sep 2017.

[71] L. von Stumberg, V. Usenko, J. Engel, J. Stueckler, and D. Cremers. From monocular SLAM to autonomous drone exploration. In *European Conference on Mobile Robots (ECMR)*, September 2017.

[72] T. Whelan, M. Kaess, H. Johannsson, M.F. Fallon, J.J. Leonard, and J.B. McDonald. Real-time large scale dense RGB-D SLAM with volumetric fusion. *Intl. J. of Robotics Research, IJRR*, 2014.

[73] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[74] Kejian Wu, Ahmed Ahmed, Georgios Georgiou, and Stergios Roumeliotis. A square root inverse filter for efficient vision-aided inertial navigation on mobile devices. 07 2015.

[75] Jean yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[76] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:13301334, December 2000.

APPENDICES

Appendix A

Visual Tracking Jacobian

The Jacobian of last iteration and is calculated using the chain rule with respect to 6 components of the pose update vector $\Delta\xi$ in the $\mathfrak{se}(3)$. The Jacobian calculated with an unprojected point $\mathbf{p} = \pi^{-1}(\tilde{\mathbf{x}}, \frac{1}{D_{T(\tilde{\mathbf{x}})}})$

$$\mathbf{J_x} = \nabla I|_{\omega(\mathbf{x},\mathbf{T_i})} \quad \left.\frac{\partial \pi}{\partial \mathbf{p}}\right|_{\mathbf{T_i p}} \quad \left.\frac{\partial \mathbf{Tp}}{\partial \mathbf{T}}\right|_{\mathbf{T_i}} \quad \left.\frac{\partial \mathbf{TT_i}}{\partial \mathbf{T}}\right|_{\mathbf{I}} \quad \left.\frac{\partial \exp(\hat{\xi})}{\partial \xi}\right|_{0} \tag{A.1}$$

These Jacobian can be informally understood as:

- $\nabla I|_{\omega(\mathbf{x},\mathbf{T_i})}$: Derivative of the new image at the warped pixel position
- $\left.\frac{\partial \pi}{\partial \mathbf{p}}\right|_{\mathbf{T_i p}}$: Derivative of the projection function at the 3D point transformed with the current pose estimate $\mathbf{T_i p}$
- $\left.\frac{\partial \mathbf{Tp}}{\partial \mathbf{T}}\right|_{\mathbf{T_i}}$: Derivative of the matrix vector multiplication of rigid body transformation at the current pose estimate with the un-projected pixel position
- $\left.\frac{\partial \mathbf{TT_i}}{\partial \mathbf{T}}\right|_{\mathbf{I}}$: Derivative of the rigid body transformation concatenation at the identity with the current pose estimate.
- $\left.\frac{\partial \exp(\hat{\xi})}{\partial \xi}\right|_{0}$: Derivative of the exp-hat function at the zero vector(corresponding to identity).

And these Jacobian evaluate to:

$$\nabla I|_{\omega(\mathbf{x},\mathbf{T_i})} = (\nabla I_x, \nabla I_y) \tag{A.2}$$

$$\frac{\partial \pi}{\partial \mathbf{p}}\Big|_{\mathbf{T_i p}} = \begin{pmatrix} f_x \frac{1}{z'} & 0 & -f_x \frac{x'}{z'^2} \\ 0 & f_y \frac{1}{z'} & -f_y \frac{y'}{z'^2} \end{pmatrix} \tag{A.3}$$

$$\frac{\partial \mathbf{Tp}}{\partial \mathbf{T}}\Big|_{\mathbf{T_i}} = \begin{pmatrix} x & 0 & 0 & y & 0 & 0 & z & 0 & 0 & 1 & 0 & 0 \\ 0 & x & 0 & 0 & y & 0 & 0 & z & 0 & 0 & 1 & 0 \\ 0 & 0 & x & 0 & 0 & y & 0 & 0 & z & 0 & 0 & 1 \end{pmatrix} \tag{A.4}$$

$$\frac{\partial \mathbf{TT_i}}{\partial \mathbf{T}}\bigg|_{\mathbf{I}} = \begin{pmatrix} r_{11} & 0 & 0 & r_{21} & 0 & 0 & r_{31} & 0 & 0 & 0 & 0 & 0 \\ 0 & r_{11} & 0 & 0 & r_{21} & 0 & 0 & r_{31} & 0 & 0 & 0 & 0 \\ 0 & 0 & r_{11} & 0 & 0 & r_{21} & 0 & 0 & r_{31} & 0 & 0 & 0 \\ r_{12} & 0 & 0 & r_{22} & 0 & 0 & r_{32} & 0 & 0 & 0 & 0 & 0 \\ 0 & r_{12} & 0 & 0 & r_{22} & 0 & 0 & r_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & r_{12} & 0 & 0 & r_{22} & 0 & 0 & r_{32} & 0 & 0 & 0 \\ r_{13} & 0 & 0 & r_{23} & 0 & 0 & r_{33} & 0 & 0 & 0 & 0 & 0 \\ 0 & r_{13} & 0 & 0 & r_{23} & 0 & 0 & r_{33} & 0 & 0 & 0 & 0 \\ 0 & 0 & r_{13} & 0 & 0 & r_{23} & 0 & 0 & r_{33} & 0 & 0 & 0 \\ t_x & 0 & 0 & t_y & 0 & 0 & t_z & 0 & 0 & 1 & 0 & 0 \\ 0 & t_x & 0 & 0 & t_y & 0 & 0 & t_z & 0 & 0 & 1 & 0 \\ 0 & 0 & t_x & 0 & 0 & t_y & 0 & 0 & t_z & 0 & 0 & 1 \end{pmatrix} \tag{A.5}$$

$$\frac{\partial \exp(\hat{\xi})}{\partial \xi}\bigg|_{0} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{A.6}$$

The final results obtained by matrix multiplication and simplification (e.g. by transforming $r_1 1x + r_1 2y + r_1 3z + t_x = x'$) is:

$$\mathbf{J_x} = \frac{1}{z'} \left(\nabla I_x f_x, \nabla I_y f_y\right) \begin{pmatrix} 1 & 0 & -\frac{x'}{z'} & -\frac{x'y'}{z'} & z' + \frac{x'^2}{z'} & -y' \\ 0 & 1 & -\frac{y'}{z'} & -z' - \frac{y'^2}{z'} & \frac{x'y'}{z'} & x' \end{pmatrix} \tag{A.7}$$

For taking the depth noise into account, the derivative of the residual with regard to the pixel's estimated inverse depth at its current estimate is required:

$$\left.\frac{\partial r_x(\mathbf{T})}{\partial D_T}\right|_{D_T(\mathbf{x})} = \nabla I|_{\omega(\mathbf{x},\mathbf{T_i})} \left.\frac{\partial \pi}{\partial \mathbf{p}}\right|_{\mathbf{T_i p}} \left.\frac{\partial \mathbf{T_i p}}{\partial \mathbf{T}}\right|_{\mathbf{p}} \left.\frac{\partial \pi^{-1}}{\partial Z}\right|_{\frac{1}{D_T(\mathbf{x})}} \left.\frac{\partial \frac{1}{x}}{\partial x}\right|_{D_T(\mathbf{x})} \tag{A.8}$$

Hence, the new Jacobians are:

- $\left.\frac{\partial \mathbf{T_i p}}{\partial \mathbf{T}}\right|_{\mathbf{p}}$: Derivative of the matrix-vector multiplication of rigid body transformations at the un-projected point with the current pose estimate.
- $\left.\frac{\partial \pi^{-1}}{\partial Z}\right|_{\frac{1}{D_T(\mathbf{x})}}$: Derivative of the un-projection function at the estimated depth with the current pixel position
- $\left.\frac{\partial \frac{1}{x}}{\partial x}\right|_{D_T(\mathbf{x})}$: Derivative of the depth inversion at the current depth estimate

The pixel coordinates are denoted p_x and p_y. As above, x, y and $z = \frac{1}{D_T}$ respectively x', y' and z' denote the components of the un-projected pixel before and after the transformation. The inverse intrinsic camera parameters $\mathbf{K}^{-1}$ is represented in the matrix form as :

$$\mathbf{K}^{-1} = \begin{pmatrix} k_{11} & 0 & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & 0 & 1 \end{pmatrix} \tag{A.9}$$

The Jacobians then evaluate to:

$$\left.\frac{\partial \mathbf{T_i p}}{\partial \mathbf{T}}\right|_{\mathbf{p}} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{A.10}$$

$$\left.\frac{\partial \pi^{-1}}{\partial Z}\right|_{\frac{1}{D_T(\mathbf{x})}} = \begin{pmatrix} k_{11}p_z + k_{13} \\ k_{21}p_y + k_{23} \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} D_T(\mathbf{x}) \tag{A.11}$$

$$\left.\frac{\partial \frac{1}{x}}{\partial x}\right|_{D_T(\mathbf{x})} = \frac{-1}{D_T(\mathbf{x})^2} \tag{A.12}$$

The final result obtained by matrix multiplication and simplification (e.g. $r_{11}x + r_{12}y + r_{13}z = x' - t_x$) is:

$$\left.\frac{\partial r_x(\mathbf{T})}{\partial D_T}\right|_{D_T(\mathbf{x})} = \frac{1}{D_T(\mathbf{x})z'^2}(\nabla I_x f_x(t_x z' - t_z x') + (\nabla I_y f_y(t_y z' - t_z y')) \tag{A.13}$$

Appendix B

Epipolar Jacobian

The epipolar residual r_{epl} is defined in (5.5) as:

$$r_{epl} = dist(\mathbf{x_{LK}^T}, \hat{\mathbf{l}}')$$

The epipolar line due to the initial pose prediction (5.4) from IMU is :

$$\hat{\mathbf{l}}' = \hat{\mathbf{F}}_{\mathbf{IMU}}\mathbf{x}$$

On expansion of (5.4) ;

$$\begin{pmatrix} a_l \\ b_l \\ c_l \end{pmatrix} = \begin{pmatrix} (\frac{1}{f_x})^2[-t_z r_{21} + t_y r_{31}] & (\frac{1}{f_x f_y})[-t_z r_{22} + t_y r_{32}] & (\frac{1}{fx})[-t_z r_{23} + t_y r_{33}] \\ (\frac{1}{f_x f_y})[t_z r_{11} - t_x r_{31}] & (\frac{1}{f_y})^2[t_z r_{12} - t_x r_{32}] & (\frac{1}{fy})[t_z r_{13} - t_x r_{33}] \\ (\frac{1}{f_x})[-t_y r_{11} + t_x r_{21}] & (\frac{1}{f_y})[-t_y r_{12} + t_x r_{22}] & [-t_y r_{13} + t_x r_{23}] \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{B.1}$$

where $(x, y, 1)$ is the normalized pixel coordinate of the original key-frame image (with only subtraction of c_x, c_y for simplicity). (a_l, b_l, c_l) is the epipolar line vector. f_x, f_y are the camera focal parameters obtained through calibration. The parameters t_i and r_{ij} denote translation and rotational elements of the pose matrix $\mathbf{T} \in \mathbf{SE(3)}$ and i, j are the indices of the matrix.

By substitution of (B.1) in (5.4)

$$r_{epl} = \frac{a_l u_{LK} + b_l v_{LK} + c_l}{\sqrt{a_l^2 + b_l^2}} \tag{B.2}$$

Its worthwhile to note here that a_l, b_l, c_l are all functions of r_{ij} and t_i which are elements of the matrix representing the rigid body transformation $\mathbf{T} \in \mathbf{SE(3)}$

For the sake of notational simplicity , r_{epl} is represented as:

$$r_{epl} = \frac{f(\mathbf{T})}{g(\mathbf{T})} \tag{B.3}$$

In order to compute Jacobian of r_{epl}, chain rule and substitution is applied.

$$\begin{aligned}
\left.\frac{\partial r_{epl}}{\partial \mathbf{T}}\right|_{\mathbf{T_i}} &= \frac{f'(\mathbf{T})g(\mathbf{T}) - g'(\mathbf{T})f(\mathbf{T})}{g(\mathbf{T})^2} && \text{(B.4)} \\
&= \frac{f'(\mathbf{T})}{g(\mathbf{T})} - \frac{g'(\mathbf{T})}{g(\mathbf{T})} r_{epl} \\
&= \frac{f'(\mathbf{T})}{g(\mathbf{T})} - \frac{r_{epl}}{g(\mathbf{T})} g'(\mathbf{T}) \\
&= \frac{f'(\mathbf{T})}{g(\mathbf{T})} - \frac{r_{epl}}{g(\mathbf{T})} [\frac{\partial g(\mathbf{T})}{\partial (a_l^2 + b_l^2)} \frac{\partial (a_l^2 + b_l^2)}{\partial \mathbf{T}}] \\
&= \frac{f'(\mathbf{T})}{g(\mathbf{T})} - \frac{r_{epl}}{g(\mathbf{T})} [-\frac{1}{2g(\mathbf{T})} (2a_l \frac{\partial a_l}{\partial \mathbf{T}} + 2b_l \frac{\partial b_l}{\partial \mathbf{T}})] \\
&= \frac{J_{f(\mathbf{T})}}{g(\mathbf{T})} + \frac{r_{epl}}{g(\mathbf{T})} (a_l J_{a(\mathbf{T})} + b_l J_{b(\mathbf{T})}) && \text{(B.5)}
\end{aligned}$$

All the variables in (B.5) are known expect the Jacobians $J_{f(\mathbf{T})}$, $J_{a(\mathbf{T})}$ and $J_{b(\mathbf{T})}$ which are stated below:

$$J_{f(\mathbf{T})} = \begin{pmatrix} v_{LK}(\frac{t_z}{f_x f_y}x - \frac{t_y}{f_x}x) \\ -u_{LK}(\frac{t_z}{f_x^2}x + \frac{t_x}{f_x}x) \\ u_{LK}(\frac{t_y}{f_x^2}x) - v_{LK}(\frac{t_x}{f_x f_y}x) \\ v_{LK}(\frac{t_z}{f_y^2}y - \frac{t_y}{f_y}y) \\ -u_{LK}(\frac{t_z}{f_x f_y}y + \frac{t_x}{f_y}y) \\ u_{LK}(\frac{t_y}{f_x f_y}y) - v_{LK}(\frac{t_x}{f_y^2}y) \\ v_{LK}(\frac{t_z}{f_y} - t_y) \\ -u_{LK}(\frac{t_z}{f_x} + t_x) \\ u_{LK}(\frac{t_y}{f_x}) - v_{LK}(\frac{t_x}{f_y}) \\ -v_{LK}(\frac{r_{31}}{f_x f_y})x - v_{LK}(\frac{r_{32}}{f_y^2})y - v_{LK}(\frac{r_{33}}{f_y})x + (\frac{r_{21}}{f_x})x + (\frac{r_{22}}{f_y})y + r_{23} \\ u_{LK}(\frac{r_{31}}{f_x^2})x + u_{LK}(\frac{r_{32}}{f_x f_y})y + u_{LK}(\frac{r_{33}}{f_y})x - (\frac{r_{11}}{f_x})x - (\frac{r_{12}}{f_y})y - r_{13} \\ -u_{LK}(\frac{r_{21}}{f_x^2})x - u_{LK}(\frac{r_{22}}{f_x f_y})y - u_{LK}(\frac{r_{23}}{f_x})x + v_{LK}(\frac{r_{11}}{f_x f_y})x + v_{LK}(\frac{r_{12}}{f_y^2})y + \frac{r_{13}}{f_y} \end{pmatrix}^{\mathbf{T}} \tag{B.6}$$

where (u_{LK}, v_{LK}) are elements of $\mathbf{x_{LK}}$.

$$J_{a(\mathbf{T})} = \begin{pmatrix} 0 \\ -\frac{t_z}{f_x^2}x \\ \frac{t_y}{f_x^2}x \\ 0 \\ -\frac{t_z}{f_x f_y}y \\ \frac{t_y}{f_x f_y}y \\ 0 \\ -\frac{t_z}{f_x} \\ \frac{t_y}{f_x} \\ 0 \\ \frac{r_{31}}{f_x^2}x + \frac{r_{32}}{f_x f_y}y + \frac{r_{33}}{f_x} \\ -\frac{r_{21}}{f_x^2}x - \frac{r_{22}}{f_x f_y}y - \frac{r_{23}}{f_x} \end{pmatrix}^{\mathbf{T}} \tag{B.7}$$

$$J_{b(\mathbf{T})} = \begin{pmatrix} \frac{t_z}{f_x f_y} x \\ 0 \\ -\frac{t_x}{f_x f_y} x \\ \frac{t_z}{f_y^2} y \\ 0 \\ -\frac{t_x}{f_y^2} y \\ \frac{t_z}{f_y} \\ 0 \\ -\frac{t_x}{f_y} \\ -\frac{r_{31}}{f_x f_y} x - \frac{r_{32}}{f_y^2} y - \frac{r_{33}}{f_y} \\ 0 \\ \frac{r_{11}}{f_x f_y} x + \frac{r_{12}}{f_y^2} y - \frac{r_{13}}{f_y} \end{pmatrix}^{\mathbf{T}} \tag{B.8}$$

Similar to A.1, the complete Jacobian with respect to pose updates in Lie Algebra $\mathfrak{se}(3)$ is obtained by chain rule.

$$\mathbf{J_{epl}} = \left.\frac{\partial r_{epl}}{\partial \mathbf{T}}\right|_{\mathbf{T_i}} \left.\frac{\partial \mathbf{T T_i}}{\partial \mathbf{T}}\right|_{\mathbf{I}} \left.\frac{\partial \exp(\hat{\xi})}{\partial \xi}\right|_{0} \tag{B.9}$$

where $\left.\frac{\partial \mathbf{T T_i}}{\partial \mathbf{T}}\right|_{\mathbf{I}}$ and $\left.\frac{\partial \exp(\hat{\xi})}{\partial \xi}\right|_{0}$ are same as that stated in (A.5) and (A.6)

In order to calculate the weights, the Jacobian $\frac{\partial \mathbf{r_{epl}}}{\partial D_i}$ is required which is:

$$\frac{\partial \mathbf{r_{epl}}}{\partial D_i} = \frac{a_l}{g(\mathbf{T})} \frac{\partial u_{LK}}{\partial D_i} + \frac{b_l}{g(\mathbf{T})} \frac{\partial v_{LK}}{\partial D_i} \tag{B.10}$$

where $g(\mathbf{T})$ is defined in (B.3) and

$$\frac{\partial u_{LK}}{\partial D_i} = \frac{t_x f_x - t_z u_{LK}}{(r_{31}\frac{x}{f_x} + r_{32}\frac{y}{f_y} + r_{33}) + t_z \hat{D}_i} \tag{B.11}$$

$$\frac{\partial v_{LK}}{\partial D_i} = \frac{t_y f_y - t_z v_{LK}}{(r_{31}\frac{x}{f_x} + r_{32}\frac{y}{f_y} + r_{33}) + t_z \hat{D}_i} \tag{B.12}$$

Appendix C

Inverse Depth Jacobian

The epipolar residual r_{D_i} is defined in (5.6) as:

$$\begin{aligned} r_{D_i} =& \hat{D}_i - g(\hat{D}_i, \mathbf{T}) \\ =& \hat{D}_i - (\mathbf{R}_{row3} \bullet \mathbf{K}\mathbf{x} + t_z) \\ =& \hat{D}_i - \frac{\hat{D}_i}{(r_{31}\frac{x}{f_x} + r_{32}\frac{y}{f_y} + r_{33}) + t_z\hat{D}_i} \end{aligned} \tag{C.1}$$

The corresponding inverse depth Jacobian using chain rule is obtained as:

$$\mathbf{J_{r_{D_i}}} = \left.\frac{\partial r_{D_i}}{\partial \mathbf{T}}\right|_{\mathbf{T_i}} \left.\frac{\partial \mathbf{T}\mathbf{T_i}}{\partial \mathbf{T}}\right|_{\mathbf{I}} \left.\frac{\partial \exp(\hat{\xi})}{\partial \xi}\right|_{0} \tag{C.2}$$

where $\frac{\partial r_{D_i}}{\partial \mathbf{T}}\Big|_{\mathbf{T_i}}$ is:

$$\frac{\partial r_{D_i}}{\partial \mathbf{T}}\Big|_{\mathbf{T_i}} = \begin{pmatrix} 0 \\ 0 \\ -\frac{g(\hat{D}_i,\mathbf{T})}{(r_{31}\frac{x}{f_x}+r_{32}\frac{y}{f_y}+r_{33})+t_z\hat{D}_i}(\frac{x}{fx}) \\ 0 \\ 0 \\ -\frac{g(\hat{D}_i,\mathbf{T})}{(r_{31}\frac{x}{f_x}+r_{32}\frac{y}{f_y}+r_{33})+t_z\hat{D}_i}(\frac{y}{fy}) \\ 0 \\ 0 \\ -\frac{g(\hat{D}_i,\mathbf{T})}{(r_{31}\frac{x}{f_x}+r_{32}\frac{y}{f_y}+r_{33})+t_z\hat{D}_i} \\ 0 \\ 0 \\ -\frac{g(\hat{D}_i,\mathbf{T})}{(r_{31}\frac{x}{f_x}+r_{32}\frac{y}{f_y}+r_{33})+t_z\hat{D}_i}(\hat{D}_i) \end{pmatrix}^{\mathbf{T}} \tag{C.3}$$

where $\frac{\partial \mathbf{T}\mathbf{T_i}}{\partial \mathbf{T}}\Big|_{\mathbf{I}}$ and $\frac{\partial \exp(\hat{\xi})}{\partial \xi}\Big|_{0}$ are same as that stated in (A.5) and (A.6)

In order to calculate the weights, the Jacobian $\frac{\partial \mathbf{r_{D_i}}}{\partial D_i}$ is required which is:

$$\frac{\partial r_{D_i}}{\partial D_i} = 1 - \frac{1}{(r_{31}\frac{x}{f_x}+r_{32}\frac{y}{f_y}+r_{33})+t_z\hat{D}_i} + \frac{t_z}{(r_{31}\frac{x}{f_x}+r_{32}\frac{y}{f_y}+r_{33})+t_z\hat{D}_i} \tag{C.4}$$