공학석사 학위논문

# Self-Calibrated Visual-Inertial Odometry for Rover Localization

## 로버 항법을 위한 자가보정 영상관성 오도메트리

2018 년 12 월

서울대학교 대학원

기계항공공학부

정 재 형

# Self-Calibrated Visual-Inertial Odometry for Rover Localization

## 로버 항법을 위한 자가보정 영상관성 오도메트리

지도교수 박 찬 국

이 논문을 공학석사 학위논문으로 제출함

2018 년 12 월

서울대학교 대학원

기계항공공학부

정 재 형

정 재 형의 공학석사 학위논문을 인준함

2018 년 12 월

위 원 장 김 유 단 (인)

부위원장 박 찬 국 (인)

위 원 김 현 진 (인)

## Abstract

# Self-Calibrated Visual-Inertial Odometry for Rover Localization

Jae Hyung Jung
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

This master's thesis presents a direct visual odometry robust to illumination changes and a self-calibrated visual-inertial odometry for a rover localization using an IMU and a stereo camera. Most of the previous vision-based localization algorithms are vulnerable to sudden brightness changes due to strong sunlight or a variance of the exposure time, that violates Lambertian surface assumption. Meanwhile, to decrease the error accumulation of a visual odometry, an IMU can be employed to fill gaps between successive images. However, extrinsic parameters for a visual-inertial system should be computed precisely since they play an important role in making a bridge between the visual and inertial coordinate frames, spatially as well as temporally. This thesis proposes a bucketed illumination model to account for partial and global illumination changes along with a framework of a direct visual odometry for a rover localization. Furthermore, this study presents a self-calibrated visual-inertial odometry in which the time-offset and relative pose of an IMU and a stereo camera are estimated by using point feature measurements. Specifically, based on the extended Kalman filter pose estimator, the calibration parameters are augmented in the

filter state. The proposed visual odometry is evaluated through the open source dataset where images are captured in a Lunar-like environment. In addition to this, we design a rover using commercially available sensors, and a field testing of the rover confirms that the self-calibrated visual-inertial odometry decreases a localization error in terms of a return position by 76.4% when compared to the visual-inertial odometry without the self-calibration.

**Keywords**: Rover localization, Direct visual odometry, Visual-inertial navigation, Self-calibration

**Student Number**: 2017-25371

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation and background

Estimating an ego-motion of a camera has been one of the most challenging tasks for a camera mounted moving platform in global navigation satellite system (GNSS) denied environment. One way to tackle this issue is to use visual odometry (VO) which was coined its name owing to its similarity to wheel odometry (WO). VO estimates a relative 6-DOF pose between consecutive images and incrementally obtains its pose and does not suffer from error accumulation caused by wheel slips, a tremendous disadvantage in WO [34]. VO is widely used in a robot navigation because of cost and space effectiveness of cameras. VO system was successfully implemented in NASA's Mars Exploration Rover (MER) and Mars Science Laboratory Curiosity rover that is shown in Fig. 1.1 [1]. MER's VO system tracked corner features at Martian terrain and estimated relative poses between an incoming pair of images by the stereo camera [3]. VO in a micro aerial vehicle (MAV) application can be found in [6] which exploited VO with a downward-looking camera attached to the MAV as shown in Fig. 1.2.

VO can be divided into two types of so-called indirect VO (IVO) and direct VO (DVO), depending on which information is provided to the cost function in the optimization problem. IVO [31] minimizes a reprojection error defined

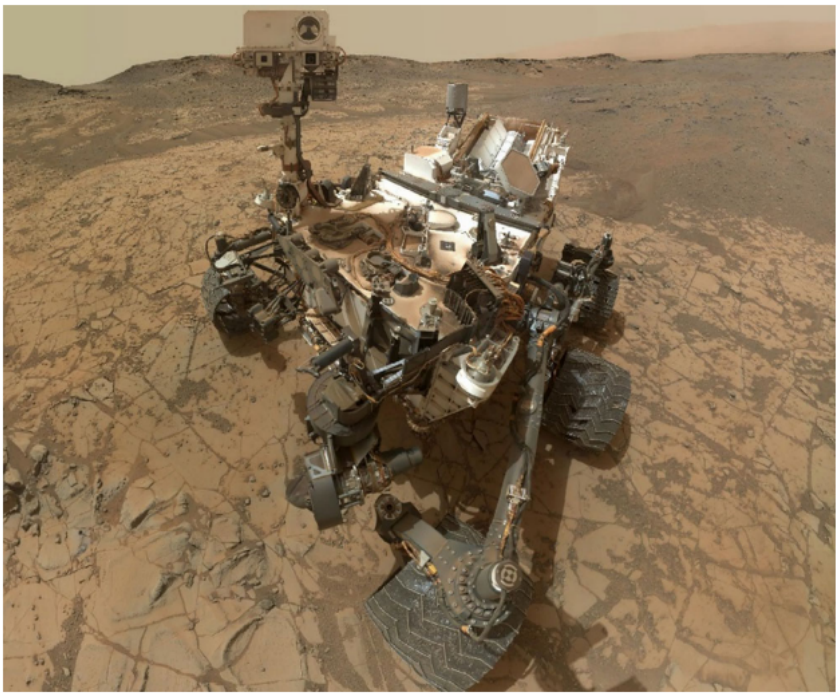


Figure 1.1: Selfie by Curiosity on sols 868 to 884 [1]

as a difference between a feature measurement and an estimated feature location, while DVO [37] estimates camera's pose by minimizing a photometric error which is intensity difference among consecutive images. DVO is known to outperform IVO in motion blurred and featureless condition since it does not utilize feature information but pixel intensities directly in images [37]. However, DVO has a substantial weakness that it is vulnerable to illumination changes in a sequence of images. This is because DVO assumes that every object in the world has the same intensity regardless of the viewer's position that is known as the Lambertian surface. The assumption is invalid under practical conditions where sudden and irregular illumination changes are prevalent in sequences of images attributable to automatic exposure and gain of a camera or albedo

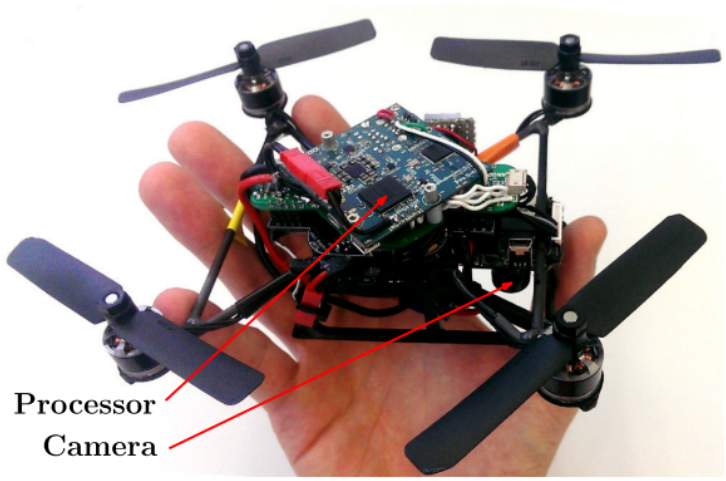



Figure 1.2: Micro aerial vehicle equipped with a downward-looking camera [6]

change that is caused by an irregular reflection under outdoor sunlight.

On the other hand, while VO suffers from the well-known error accumulation, visual-inertial odometry (VIO) decreases its rate by filling a gap between images using inertial measurement unit (IMU) readings [32]. A fusion of a camera and IMU is an attractive solution due to their complementary features; a camera provides rich information for a localization with low sampling time, measurements of an IMU give the absolute scale with fast sampling time. Moreover, measurements from a camera provide constraints to a pose of a sensing platform that reduces unbounded error accumulation caused by consumer-grade IMU.

Most of the visual-inertial fusion algorithms assume that output data from a camera and IMU is timely synchronized and the sensors are spatially well aligned. However, this causes significant estimation errors when time-delay of a camera is not negligible or a camera-IMU system is not well calibrated since the measurement model is linearized around the currently available estimate

referenced at the camera frame. Even if a camera-IMU system is calibrated in advance, this cannot reflect uncertainties on calibration parameters to an estimator. In the worst case, calibration parameters could be changed due to external shocks.

In this paper, we focus on the problem of how to deal with sudden illumination changes in DVO, and extrinsic calibration parameter issues in a visual-inertial system. For the first topic, we propose a strategy called *bucketed local illumination model* to effectively model brightness changes in captured images. For the second topic, we exploit the theoretic results of [11], [25] and formulate a self-calibrated extended Kalman filter (EKF)-based VIO algorithm using feature point measurements obtained from the stereo camera.

## 1.2 Objectives and contributions

The objectives of this study are designing a DVO that is robust to illumination changes and a self-calibration of a visual-inertial system using point measurements using a stereo camera. The main contributions of this thesis are:

- we propose a patch-based DVO which is robust to illumination changes at stereo camera images employing the *bucketed local illumination model*. In our model, patches centered at feature points have the same affine illumination parameters within the buckets in the image. Therefore, the proposed model requires less computational cost than the local illumination model which augments its state vector per a patch. Also, the generated patches enable the proposed method to work in a more general environment, since it does not require any artificial planar patches while accounting for not only global light changes but also local light changes.

- we propose the *adaptive prior weight* as a function of the previously con-

verged motion in a constant velocity motion model framework. This reflects a physical intuition that the faster a camera moves, the harder it is to change a velocity—we assign a weight to the constant velocity model according to the previous motion. In cases where a motion is huge, the proposed method improves estimation accuracy, as will be seen in Chapter 3.

- we show experimental evidence that the proposed algorithm outperforms global illumination model in the lunar-like terrain dataset [8] with strong outdoor sunlight and the MAV dataset [2] where camera's automatic exposure and gain make sudden and partial illumination changes throughout image sequences.

- Self-calibration VIO is formulated using point measurements from a stereo camera. Specifically, temporal and spatial extrinsic parameters of a visual-inertial system are augmented in the filter state in an EKF-framework. We experimentally prove that the calibration parameters play an important role when fusing visual-inertial sensors. We design a testing rover integrated into Robot Operating System (ROS) using a commercially available platform and sensors.

The remainder of this study is organized as follows. In Chapter 2, we introduce related literatures of VO and VIO. Since a scope of VO and VIO is too tremendous to cover in this study, we focus on previous works dealing with brightness change issues and estimating calibration parameters in online. Next, in Chapter 3, the proposed VO algorithm is presented with its mathematical derivation and experimental results. Chapter 4 describes the self-calibrated visual-inertial algorithm using an IMU and a stereo camera, and the rover field testing. Finally, Chapter 5 summarizes the conclusion of this study.

# Chapter 2

# Related Works

In this chapter, we begin with related works on VO. The famous VO algorithms will be introduced, then previous studies dealing with sudden brightness changes due to strong outdoor sunlight or camera exposure settings especially for the context of a localization will be reviewed. In what follows, methods fusing measurements from an IMU and camera will be briefly discussed, and previous literatures on a self-calibration of the visual-inertial system will be reviewed.

## 2.1 Visual odometry

Visual Odometry (VO) minimizes either a reprojection error of tracked feature points or a photometric error that is not a geometrical distance but an image intensity. The famous VO pipeline that minimizes the reprojection error is the work of Geiger et al. [10], and its predecessor, [19] by Kitt et al.

In the work of [19], an iterated sigma point Kalman filter (ISPKF) using a constant velocity model and a trifocal tensor is proposed. As shown in Fig. 2.1, the preprocessed feature points are fed into the filter where the trifocal tensor and the constant velocity model is fused in an optimal sense. The trifocal tensor encapsulates projective geometry between different view points excluding feature information in the filter state (structureless). In addition to the proposed

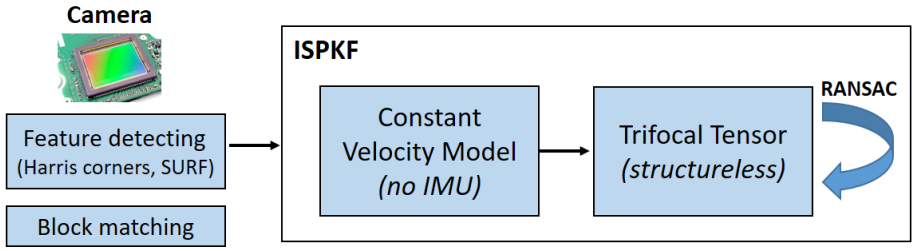



Figure 2.1: Visual odometry pipeline proposed by Kitt et al. [19]

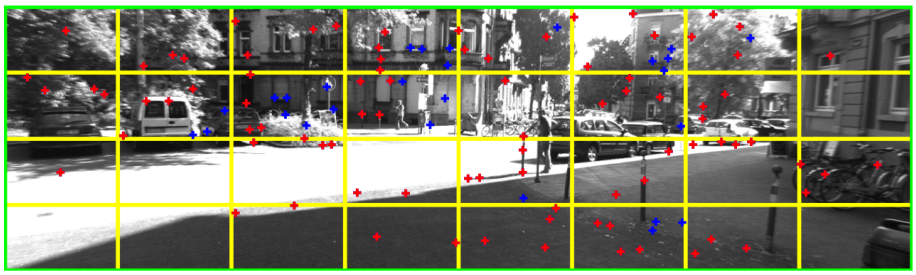


Figure 2.2: Sample image of bucketing mechanism [19]

estimator, they introduce a feature selection strategy called Bucketing shown in Fig. 2.2. The grids formed by yellow lines in Fig. 2.2 are buckets and the certain number of features are distributed in the yellow boxes. It is reported that a uniform distribution of feature plays an important role in a localization problem using images.

The work by Forster et al. [6, 7] exploits both geometry and intensity information of feature points. The flowchart and sample images are shown in Fig. 2.3. They named this pipeline as semi-direct visual odometry (SVO). This is because the direct method obtains feature correspondences, then the bundle adjustment minimizes the reprojection error in a given window. SVO aligns extracted feature patches in an image to estimate a rough initial pose and feature correspondences (Sparse Model-based Image Alignment) in Fig. 2.3(a). Then, it adjusts 2D feature locations where they adopt different techniques depending

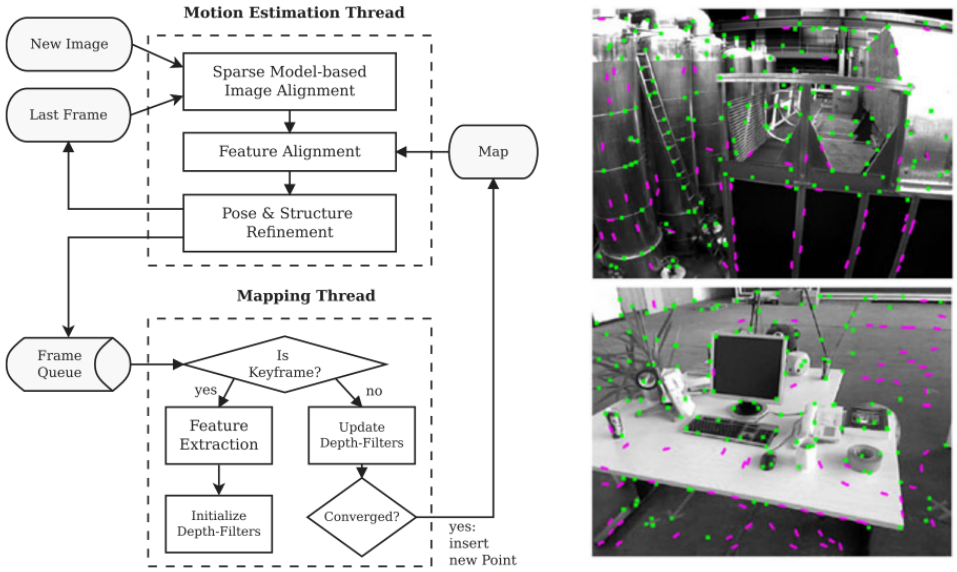



Figure 2.3: Flowchart and front-end samples of SVO [7]

on whether it is a corner or an edge (Feature Alignment) in Fig. 2.3(a). Lastly, the bundle adjustment refines feature positions and camera poses in a given window.

However, illumination issues are common in image related problems which directly exploit pixel intensities such as DVO and target tracking algorithms. Specifically, [30] employed a photometric normalization method for the face tracking algorithm under illumination changes. Also, [18] proposed the algorithm that recognizes the current illumination level of the environment and selects one of the pre-built maps with different brightness for the localization. In VO, [43] formulated the local illumination parameters for each planar patch and then marginalized out by projecting to the null space in the EKF framework. Also, [13, 20] estimated global illumination parameters with the camera's

poses employing the illumination affine model [12] and the single illumination offset, respectively from RGB-D images. In [5], the global affine illumination parameter was estimated in the alternative fashion that fixes the pose and the illumination parameter in turn to deal with the outliers. To take account of local illumination changes, [17] selected planar patches sharing the same affine illumination parameters [12] and jointly optimized the photometric error for the pose and the affine parameter in RGB-D cameras. However, the images should have enough planar patches to obtain reliable motion estimation in [17] which limits application domain into an indoor environment where artificial structures make rich planar patches for proper motion estimation.

## 2.2 Visual-inertial odometry

The method for fusing visual and inertial measurements can be broadly divided into optimization based [21, 32] and filtering-based method [28, 43]. The optimization-based algorithms minimize residuals computed from measurements of vision and IMU to obtain optimal solution, while the filtering-based algorithms sequentially update its state vector usually in extended Kalman filter (EKF) or unscented Kalman filter (UKF) framework. The filtering-based approaches can be categorized according to whether the visual feature information is included in filter's state vector: the simultaneous localization and mapping (SLAM) and the visual-inertial odometry (VIO) approach. The SLAM method [33, 36] includes feature positions in its state vector exploiting a geometric constraint with features at the current camera frame. The VIO method [23, 28], however, marginalizes feature positions in the state vector and instead possesses the history of camera poses (sliding windows) using feature measurements among multiple camera frames in the EKF-framework. Fig. 2.4 shows an

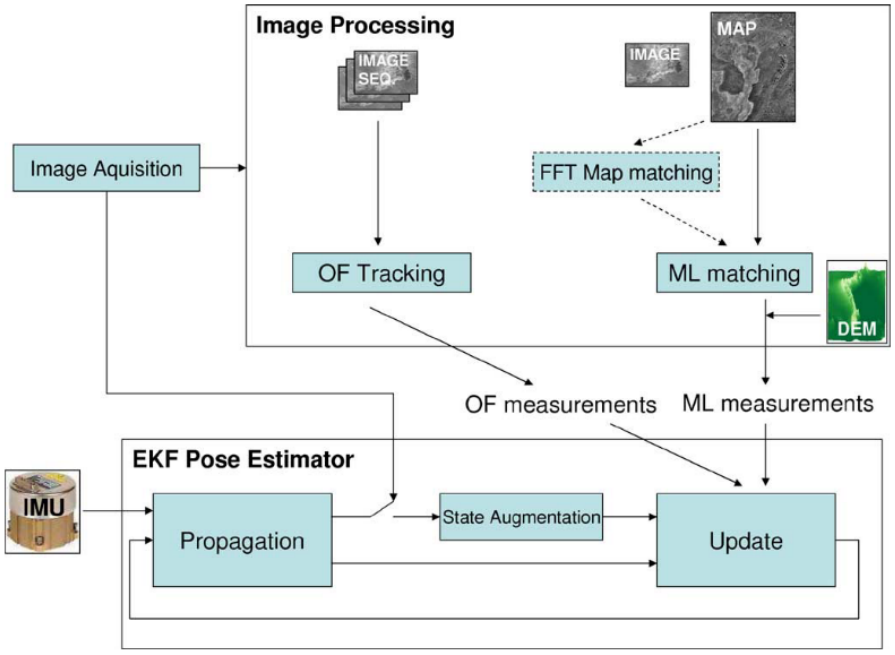



Figure 2.4: Flow chart of multi-state constraint Kalman filter (MSCKF) in Entry, Descending, and Landing mission [29]

example of a VIO implemented in an entry, descent, and landing mission.

It is known that EKF-SLAM and VIO is the optimal maximum a posterior (MAP) estimator in a case of linear Gaussian noises [23]. However, the real world is not the case, therefore, both algorithms exhibit different behavior on the same dataset. Li et al. [22, 24] proposed the method optimally utilizing SLAM/VIO algorithm (hybrid) in terms of computational cost, that is motivated by complementary computational characteristics. It is reported that utilizing information from long-observed features improve estimator's accuracy [40, 42]. In particular, Wu et al. [42] exploited long-observed features in a framework of a square root inverse filter in order to enhance pose tracking performance. However, the above-mentioned research focused on monocular vision scenario.

In contrast to the monocular vision system, the stereo system reliably initializes feature positions by virtue of the baseline between two cameras, and several pieces of research [5, 7, 39, 41] have been utilizing a stereo vision for localizing the sensing platform. Specifically, Sun et al. [39] proposed stereo camera based VIO algorithm and evaluated the algorithm on the fast flight dataset and publicly available dataset.

However, most of the previous VIOs assume that the calibration parameters are perfectly known from a calibration procedure in advance. Consequently, this cannot reflect uncertainties on calibration parameters, and in an extreme case, parameters can be changed due to external shocks. Many efforts to deal with these problems have been made. The authors of [14] showed that the IMU-cam extrinsic parameter, the scale factor, and the global gravity is observable with the global pose measurements. However, the measurement model which assumes that images output global poses was somewhat unrealistic. Guo et al. in [11] proved that cam-IMU extrinsic parameter is observable using the proposed basis functions under the known depth (feature point) assumption. The work of [25] focused on the temporal calibration of a cam-IMU system. They theoretically showed that time-offset between cam-IMU system can be recovered, while practically implemented the online calibration algorithm in the extended Kalman filter (EKF) framework. Also in [26], camera intrinsics, as well as IMU intrinsics (misalignment, g-sensitivity) was modeled in the estimator.

# Chapter 3

# Direct Visual Odometry at Outdoor

In this chapter, we present a patch-based direct visual odometry (DVO) that is robust to illumination changes at a sequence of stereo images. Illumination change violates the photo-consistency assumption and degrades the performance of DVO, thus, it should be carefully handled during minimizing the photometric error. Our approach divides an incoming image into several buckets, and patches inside each bucket own its unique affine illumination parameter to account for local illumination changes for which the global affine model fails to account, then it aligns small patches placed at temporal images. We do not distribute affine parameters to each patch since this yields huge computational load. Furthermore, we propose a prior weight as a function of the previous pose in a constant velocity model which implies that the faster a camera moves, the more likely it maintains the constant velocity model. Lastly, we show that the proposed illumination model accounts for both local and global brightness changes in synthetic image sequences. Furthermore, we verify that the proposed algorithm outperforms the global affine illumination model at the publicly available micro aerial vehicle and the planetary rover dataset which exhibit irregular and partial illumination changes due to the automatic exposure of the camera and the strong outdoor sunlight, respectively.

## 3.1 Direct visual odometry

### 3.1.1 Notations

DVO estimates relative poses between a current camera frame, $\{C_2\}$ and a previous camera frame, $\{C_1\}$ and concatenates them to obtain global poses referenced at a global frame, $\{G\}$. The relative pose, $\boldsymbol{\xi} \in se(3)$ is defined as

$$\boldsymbol{\xi} = \begin{bmatrix} {}^{C_2}\mathbf{v}_{C_1}^{\mathrm{T}} & {}^{C_2}\mathbf{w}_{C_1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \Delta t \tag{3.1}$$

where $\Delta t$ is timestamp interval between $\{C_1\}$ and $\{C_2\}$, and $\mathbf{v}$ and $\mathbf{w}$ are linear and angular velocity, respectively. Throughout this paper, the left superscript refers to a referenced frame and the right subscript refers to an object frame. Also, a bold lowercase stands for a vector, while a bold uppercase denotes a matrix. $\boldsymbol{\xi}$ is mapped to Special Euclidean group, $\mathbb{SE}(3)$ through an exponential mapping,

$${}^{C_2}\mathbf{T}_{C_1} = \exp(\hat{\boldsymbol{\xi}}) \in \mathbb{SE}(3) \tag{3.2}$$

where $\mathbf{T}$ is a rigid body transformation matrix and the hat operator, $\hat{\cdot}$ is defined as follow with the skew-symmetric matrix operator, $[\cdot_\times]$

$$\hat{\boldsymbol{\xi}} = \begin{bmatrix} \begin{bmatrix} {}^{C_2}\mathbf{w}_{C_1\times} \end{bmatrix} & {}^{C_2}\mathbf{v}_{C_1} \\ 0 & 1 \end{bmatrix} \tag{3.3}$$

### 3.1.2 Camera projection model

In this paper, the standard pinhole camera model is adopted and the projection model for a $j$-th feature viewed at $\{C_1\}$, is defined as

$$\begin{bmatrix} u_{f_j} \\ v_{f_j} \end{bmatrix} = \boldsymbol{\Pi}({}^{C_1}\mathbf{P}_{f_j}) = \begin{bmatrix} \frac{f_u \, {}^{C_1}X_{f_j}}{{}^{C_1}Z_{f_j}} + c_u \\ \frac{f_v \, {}^{C_1}Y_{f_j}}{{}^{C_1}Z_{f_j}} + c_v \end{bmatrix} \tag{3.4}$$

where ${}^{C_1}\mathbf{P}_{f_j} = \begin{bmatrix} {}^{C_1}X_{f_j} & {}^{C_1}Y_{f_j} & {}^{C_1}Z_{f_j} \end{bmatrix}^{\mathrm{T}}$ is the location of the $j$-th feature, $\mathbf{\Pi}$ is the projection model, $f_{u,v}$ and $c_{u,v}$ are a horizontal (u), and vertical (v) focal length and a principle point, respectively. A warping is a transformation of a pixel location from one image plane to another according to a relative motion between $\{C_1\}$ and $\{C_2\}$, and the warping function, $\mathbf{w}(\cdot)$ for the $i$-th pixel, $\mathbf{x}_i \in \mathbb{R}^2$ is defined as

$$\mathbf{w}(\boldsymbol{\xi}, \mathbf{x}_i) = \mathbf{\Pi}(\mathbf{g}({}^{C_2}\mathbf{T}_{C_1}(\boldsymbol{\xi}),\ \mathbf{\Pi}^{-1}(\mathbf{x}_i))) \tag{3.5}$$

where $\mathbf{g}(\mathbf{T},\ \mathbf{p}) = \mathbf{p}^{'} \in \mathbb{R}^3$ is a rigid body motion mapping. In other words, the warping is a projection of the same feature to different image planes of camera frame according to the their relative pose.

### 3.1.3 Photometric error

DVO assumes that every object in a image has Lambertian surface property, i.e. photo-consistency assumption, therefore, the photometric error for the $i$-th pixel is defined as

$$r_i(\boldsymbol{\xi}) = I_1(\mathbf{x}_i) - I_2(\mathbf{w}(\boldsymbol{\xi}, \mathbf{x}_i)) \tag{3.6}$$

where $I_1 : \mathbb{R}^2 \to \mathbb{R}$ and $I_2 : \mathbb{R}^2 \to \mathbb{R}$ are previous and current grayscale images, respectively [37]. DVO minimizes the squared sum of photometric errors with respect to the relative pose.

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{i}^{n} r_i^2(\boldsymbol{\xi}) \tag{3.7}$$

Eq. (3.7) is also known as the image alignment problem in the sense that the 6-DOF pose, $\boldsymbol{\xi}$ aligns a pair of temporal images.

## 3.2 The proposed algorithm

An overview of the proposed algorithm is presented in Fig. 3.1; (a, b) reconstruct features based on the static stereo baseline, ${}^{C_L}\mathbf{T}_{C_R}$ (e) calculate the Jacobian matrix, $\mathbf{J}_I$, residuals, $\mathbf{r}_I$, $\mathbf{r}_P$ and weighting matrices, $\mathbf{W}_I$, $\mathbf{W}_P$ according to (c) the constant velocity model with the adaptive prior weight and (d) the bucketed local illumination model under (f) the image pyramid loops. First of all, features (corner, blob, etc.) are extracted from a pair of stereo images in the bucketed manner like in [19], and a matching algorithm finds a correspondence for each feature, then, the matched features are reconstructed by two-view structure-from-motion optimization. Small patches are generated, which are centered at the extracted features. Next, the prior pose yields the prior residual and the adaptive prior weight. We augment the state vector with the affine illumination parameters and jointly estimate the relative pose and the illumination parameters based on the *bucketed local illumination model*. The current estimate of the augmented state vector iteratively computes the photometric error term. To obtain a good initial guess for the estimator, we employ the coarse-to-fine scheme as in [37]. After solving the optimization problem, the obtained relative pose is concatenated to calculate the global pose of the camera, ${}^{G}\mathbf{T}_{C_k}$. The detailed explanation for the algorithm is given in the following subsections.

### 3.2.1 Problem formulation

The main objective of this paper is to solve the photometric minimization problem, Eq. (3.8) to obtain the relative pose between two temporally successive images. However, Eq. (3.8) might converge to a false minimum or even diverge under a severe brightness change environment or a large motion of a camera. In other words, if the brightness change affects the temporal images or the overlap-

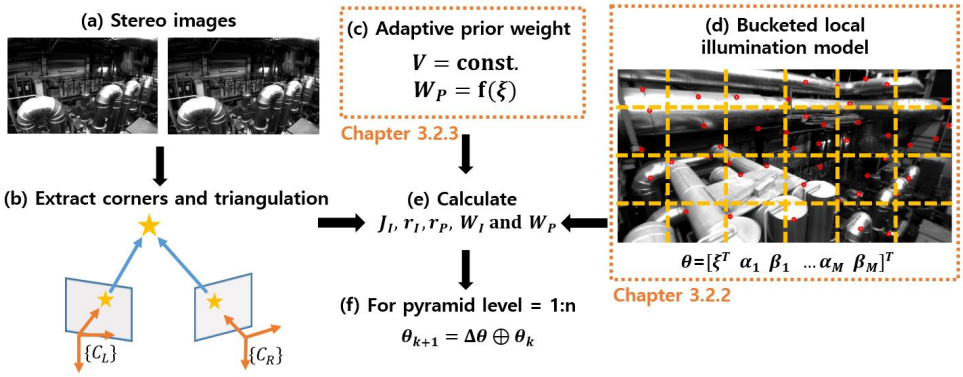



Figure 3.1: The overview of the proposed algorithm

ping region between the consecutive images is not large enough, the nonlinearity of the cost function is increased so that the estimator might fail. To account for the issues, we solve the following modified optimization problem,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \begin{bmatrix} \mathbf{r}_I^{\mathrm{T}} & \mathbf{r}_P^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{W}_I & 0 \\ 0 & \mathbf{W}_P \end{bmatrix} \begin{bmatrix} \mathbf{r}_I \\ \mathbf{r}_P \end{bmatrix} \tag{3.8}$$

that is equivalent to the maximum a posterior (MAP) estimator of $p(\boldsymbol{\xi}|\mathbf{r}, \boldsymbol{\xi}_P)$ where $p(\mathbf{r}) = p(\mathbf{r}_{1:n})$ with the independent and identically distributed (iid) assumption for the measurements and zero mean of photometric errors [16]. We denote a prior as subscript $P$ in the following sections, for instance, $\boldsymbol{\xi}_P$ stands for a pose prior. In Eq. (3.8), $\boldsymbol{\theta}$ is the state vector composed of the relative pose and affine illumination parameters defined as follow

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\xi}^{\mathrm{T}} & \alpha_1 & \beta_1 & \cdots & \alpha_M & \beta_M \end{bmatrix}^T \in \mathbb{R}^{6+2M} \tag{3.9}$$

The affine illumination parameter, $\alpha_l$ and $\beta_l$ are contrast and brightness changes, respectively [12], and $M$ stands for the total number of buckets. In Eq. (3.8), $\mathbf{W}_I$ is an image weighting matrix which is determined by the distribution of

the photometric error. For example, [16] proposed several weighting matrices such as the T-distribution weighting matrix. Also, $r_I$ is vectorized illumination compensated photometric error, and $\mathbf{r}_P$ is the residual from the prior pose,

$$\mathbf{r}_I = \begin{bmatrix} \mathbf{r}_{1,\text{affine}} & \mathbf{r}_{2,\text{affine}} & \cdots & \mathbf{r}_{n,\text{affine}} \end{bmatrix}^{\mathrm{T}} \tag{3.10}$$

$$\mathbf{r}_P = \boldsymbol{\xi}_P - \boldsymbol{\xi} \tag{3.11}$$

A choice of interesting pixels, i.e. the elements of $\mathbf{r}_I$ in Eq. (3.10) is a crucial strategy in terms of computational efficiency and estimation accuracy. For instance, [37] uses all pixels whose depth are valid for a motion tracking, and [4] reduces interesting image region to pixels with non-negligible intensity gradient. Also, [6] proposes the sparse image alignment that aligns patches centered at a sparse set of features. We adopt the sparse image alignment proposed by [6] because we do not triangulate all pixels but corner features from the static stereo pair. In addition to this, the constant depth assumption in a small patch is reasonable while reducing computational burden.

### 3.2.2 Bucketed illumination model

The number of the illumination parameters in Eq. (3.9) is directly related to the dimension of the state vector. Therefore, assigning the parameters to each pixel is computationally impractical. For instance, in a $640 \times 480$ resolution image, its state vector has 614,406 dimensions. On the other hand, a global illumination model as in [20] where $M = 1$ assumes that whole pixels in an image undergo the same intensity changes with the single pair of the parameter. However, this assumption is violated in practical applications due to partial illumination changes in the image. To address this issue, we propose a local illumination model that accounts for both global and local brightness changes.

To deal with sudden and partial illumination changes that violate the photo-consistency assumption, we distribute the unique affine illumination parameters to the sparse patches in each bucket. Note that a bucket in an image is a region divided by the grids as shown in Fig. 3.1d, for instance, $M = 24$ in case of Fig. 3.1d. Accordingly, similar to [20], the photometric error is modified as follow

$$\mathbf{r}_{i,\text{affine}}(\boldsymbol{\theta}) = I_1(\mathbf{x}_i) - [(\alpha_l + 1)I_2(\mathbf{w}(\boldsymbol{\xi}, \mathbf{x}_i)) + \beta_l] \tag{3.12}$$

and the linearized photometric error of Eq. (3.7) is

$$\mathbf{r}_I(\boldsymbol{\theta}_{k+1}) \cong \mathbf{r}_I(\boldsymbol{\theta}_k) + \mathbf{J}_I(\boldsymbol{\theta}_k)\Delta\,\boldsymbol{\theta} \tag{3.13}$$

where the augmented state vector yields the following modified Jacobian matrix,

$$\mathbf{J}_I = -\begin{bmatrix} \frac{\partial \mathbf{r}_{I,\text{affine}}}{\partial \boldsymbol{\xi}} & \frac{\partial \mathbf{r}_{I,\text{affine}}}{\partial \alpha_1} & \frac{\partial \mathbf{r}_{I,\text{affine}}}{\partial \beta_1} & \cdots & \frac{\partial \mathbf{r}_{I,\text{affine}}}{\partial \alpha_M} & \frac{\partial \mathbf{r}_{I,\text{affine}}}{\partial \beta_M} \end{bmatrix} \tag{3.14}$$

$$\frac{\partial \mathbf{r}_{I,\text{affine}}}{\partial \alpha_i} = \begin{bmatrix} 0 & \cdots & -I_2(\mathbf{w}(\boldsymbol{\xi}, \mathbf{x}_i))|_{\boldsymbol{\xi}_k} & \cdots & 0 \end{bmatrix}^{\mathrm{T}} \tag{3.15}$$

$$\frac{\partial \mathbf{r}_{I,\text{affine}}}{\partial \beta_i} = \begin{bmatrix} 0 & \cdots & -1 & \cdots & 0 \end{bmatrix}^{\mathrm{T}} \tag{3.16}$$

$$\frac{\partial \mathbf{r}_{I,\text{affine}}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} (\alpha_1 + 1) & \cdots & (\alpha_M + 1) \end{bmatrix}^T \mathbf{J}_{\boldsymbol{\xi}}(\boldsymbol{\xi}_k) \tag{3.17}$$

where $\mathbf{J}_{\boldsymbol{\xi}}$ is the Jacobian matrix of the photometric error that is computed by the chain rule from Eq. (3.5) as follows,

$$\mathbf{J}_{\boldsymbol{\xi}}(\boldsymbol{\xi}_k) = \left.\frac{\partial I_2}{\partial \Pi}\right|_{\Pi_k} \left.\frac{\partial \Pi}{\partial \mathbf{g}}\right|_{\mathbf{g}_k} \left.\frac{\partial \mathbf{g}}{\partial^{C_2}\mathbf{T}_{C_1}}\right|_{\mathbf{T}_k} \left.\frac{\partial^{C_2}\mathbf{T}_{C_1}}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}_k} \tag{3.18}$$

where $\partial I_2/\partial \Pi$ is an image gradient, $\partial \Pi/\partial \mathbf{g}$ is a derivative of a pixel position to its 3-D position, $\partial \mathbf{g}/\partial \mathbf{T}$ is a derivative of a 3-D position of a feature to a rigid body motion, and $\partial \mathbf{T}/\partial \boldsymbol{\xi}$ is a derivative of a rigid body motion to a twist.

These Jacobian matrices are derived in Appendix A in detail. Then, a normal equation is obtained after solving the first order necessary condition for Eq. (3.7),

$$\Delta\,\boldsymbol{\theta} = (\mathbf{J}_I^{\mathrm{T}}\mathbf{W}_I\mathbf{J}_I + \mathbf{W}_P)^{-1}(-\mathbf{J}_I^{\mathrm{T}}\mathbf{W}_I\mathbf{r}_I + \mathbf{W}_P\mathbf{r}_P) \tag{3.19}$$

Lastly, the relative pose is updated through exponential and logarithm mapping with the hat operator defined in Eq. (3.2),

$$\hat{\boldsymbol{\xi}}_{k+1} = \log(\exp(\Delta\,\hat{\boldsymbol{\xi}}) \cdot \exp(\hat{\boldsymbol{\xi}_k})) \tag{3.20}$$

We suppose that each patch located in the same bucket possesses its own affine parameter to account for local illumination changes, and name this model as the *bucketed local illumination model*. Note that since a pair of the parameter adds two additional states to the state vector, $\boldsymbol{\theta}$ in Eq. (3.9), we do not distribute affine parameters to each patch but to patches that belong to each bucket for reducing computational burden while accounting for local illumination changes. Also, pixels in each patch share the same parameters because of the fact that the small patches, e.g. $3 \times 3$, can be approximated locally tangent plane in a smooth surface showing similar intensity changes to illumination changes [12]. Also, in a temporal sequence of images, we suppose that patches stay within the same buckets without loss of generality because camera's frame rate (10-60 fps) is high enough to make patches stay in their bucket in general. By assigning a large number of small patches rather than few large patches as in [17], we do not need planar patch fitting and all depth value inside each patch. Also, since the proposed algorithm does not align artificial planar patches, it can operate in both outdoor and indoor environments.

### 3.2.3 Adaptive prior weight

The prior weight, $\mathbf{W}_P$ indicates how certain we believe the constant velocity model in the total cost function, Eq. (3.7), i.e. the inverse of prior's uncertainty in the MAP estimator. To deal with this weighting matrix in the absence of additional sensor like IMU or odometry, [15] conducts parametric studies and obtains the best constant diagonal weighting matrix at its given dataset in a heuristic manner. However, in this paper, we suppose the system model as the 1st order Markov model and propose the weighting matrix as a function of previously converged velocity. More specifically, the weighting matrix is calculated as follow,

$$\mathbf{W}_P = \alpha||\boldsymbol{\xi}_P||_2 I_6 \tag{3.21}$$

where $\alpha$ is a constant slope and $I_6$ is 6 by 6 identity matrix. We are motivated by the fact that the faster a camera moves, the more feasible it is affected by the previous pose because of its inertial force. Under a usual camera operation, the camera gathers images at a rate of 10-60 fps. Therefore, the time interval between incoming images is short enough to assume that the current estimator is highly influenced by how fast the previous pose was.

## 3.3 Experimental results

### 3.3.1 Synthetic image sequences

In this section, we evaluate how much illumination changes our method can tolerate in EuRoC dataset [2]. Since it is not clear to quantify the amount of illumination changes, to test our method, we follow the 2 evaluation procedures. First, we compare the pose estimation performance from the conventional method (global illumination model) and the proposed method using a pair of temporal images illuminated by randomly generated brightness changes. Second, we test our method increasing a global brightness in a pair of temporal images.

For the first evaluation, we generate images randomly illuminated by 2x2 buckets. The sample images are shown in Fig. 3.2. Please note that, we run the simulation at the 1 temporally paired image $(t_{k-1}, t_k)$, and we add the illumination change to the image at $t_k$. Since we know the exact ground truth pose at the sample pair, we can evaluate estimation performance. We ran 100 times for the given sequence with 3x3 and 5x5 buckets, and the results are plotted in Fig. 3.3. Also, we calculate the relative pose error (RPE) for each method. Note that the proposed method outperforms the conventional method at the above synthesized pair as shown in Table. 3.1.

Table 3.1: Performance comparison at the synthetic pair.

| | Conventional method | Proposed (3x3) | Proposed (5x5) |
|---|---|---|---|
| RPE RMSE [m/s] | 0.007881 | **0.001578** | **0.001578** |

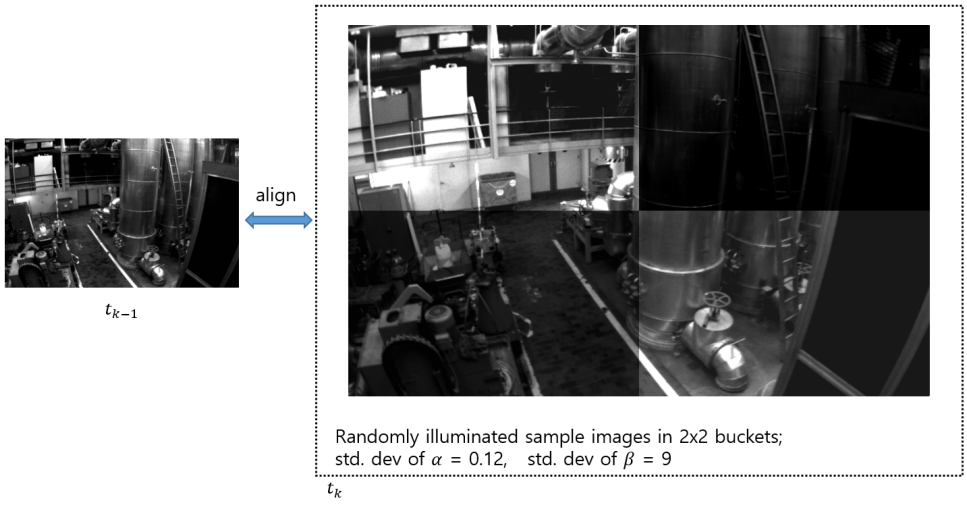



Figure 3.2: Synthetic image sequence to simulate local brightness change

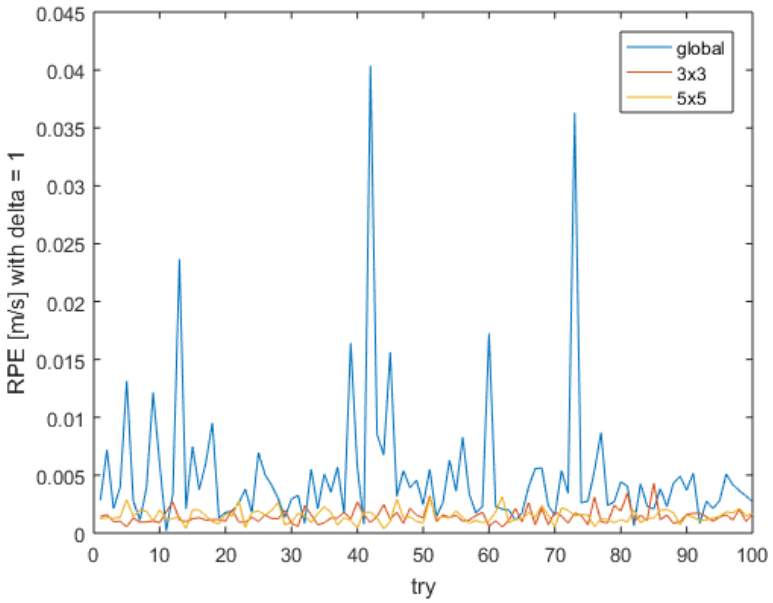



Figure 3.3: Robustness to a local brightness change

Next, we evaluate the proposed method (3x3 buckets) globally increasing intensities of the image; we generate synthetic images which contains user-defined illumination changes (deterministic). The sample images are shown in Fig. 3.4. In this setup, the illumination parameters $\alpha, \beta$ are set to 0.2g and 5g, respectively, and g is a variable that varies from 0 to 25 with 0.1 interval. Please note that, we run the simulation at the 1 temporally paired image $t_{k-1}, t_k$, and we add the illumination change to the image at $t_k$. Since we know the exact ground truth, we can evaluate how the bucketed illumination model deals with the illumination change. Fig. 3.5 shows RPE as the brightness increases. Roughly speaking, the error increases rapidly after $g = 15$.

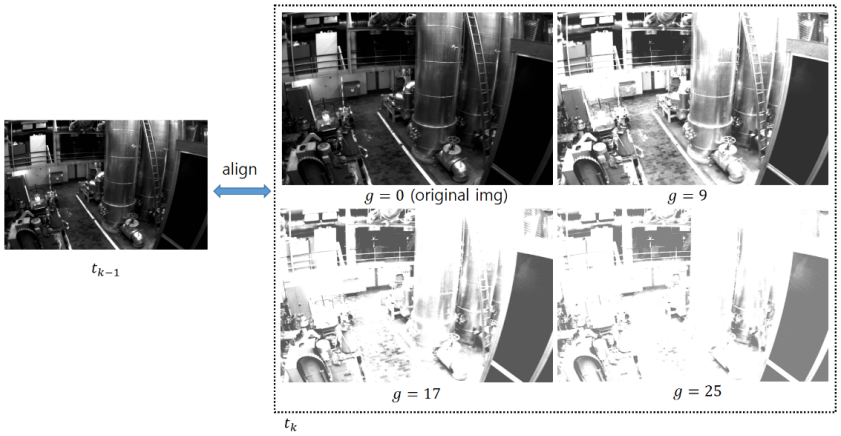



Figure 3.4: Synthetic image sequence to simulate global brightness change

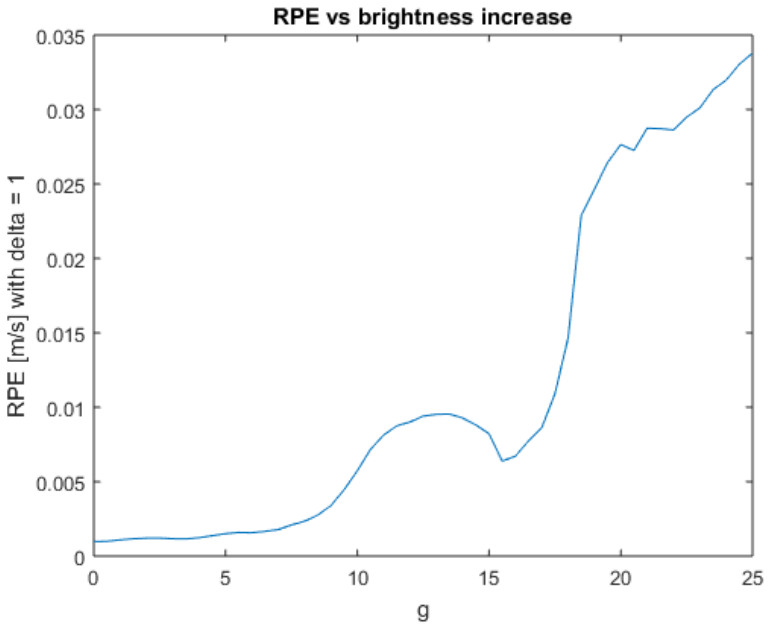



Figure 3.5: Robustness to a global brightness change

### 3.3.2 MAV datasets

The algorithm is evaluated at the real-world dataset, EuRoC dataset which is recorded by the stereo camera mounted at the MAV [2]. We have to mention that even if the image sequences are successive, there exist substantial illumination changes violating the photo-consistency assumption. Fig. 3.6a shows sample buckets extracted at the pair of temporally consecutive images with the interval of 50 ms. Specifically, $A_i$ and $B_i$ are buckets at the timestamp $t_i\,(i = 1, 2)$, for example, the pair of $A_1$, $A_2$ corresponds to the same bucket at the different instance. To verify illumination changes, we draw the intensity difference histogram for the pair of $A_i$ and $B_i$ in Fig. 3.6b. We observe that intensity differences are not negligible, also the histograms for the bucket pairs are not identical to each other in Fig. 3.6b. Therefore, to obtain reliable pose of the MAV, local and global illumination changes should be considered.

For implementation details, we obtain feature correspondences between the left and right stereo image using minimum eigenvalue feature detection [35] and Kanade Lucas Tomasi (KLT) tracker [27]. Note that the KLT tracker does not track features at temporal images but features at static stereo images where extrinsic parameter, ${}^{C_2}\mathbf{T}_{C_1}$ is calibrated in advance. Also, we maintain 100-150 number of $3 \times 3$ patches in $5 \times 5$ buckets at 20 fps image sequence, and to suppress large photometric errors, we employ T-distribution image weighting matrix as in [16]. Lastly, we iteratively solve the optimization problem using the Levenberg-Marquardt algorithm, and the proposed algorithm is implemented in MATLAB.

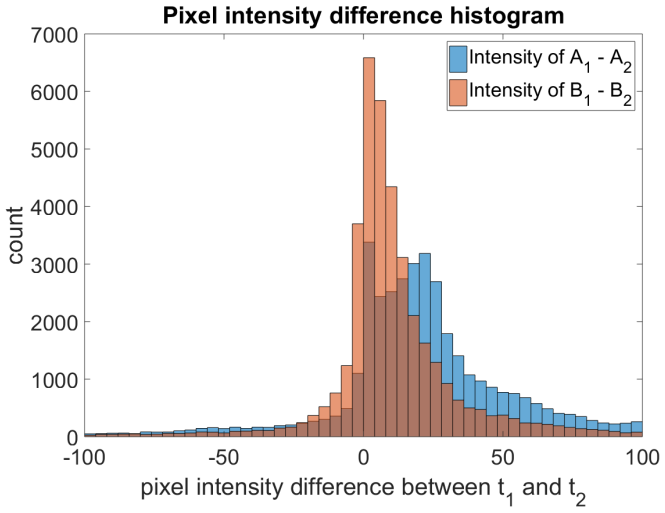
The ground truth trajectory and attitude are provided by a motion capture system, and the MAV flies 63.2-meter long trajectory for 110 seconds. We compare five different cases, i.e. sparse image alignment (sia), 'sia' with con-

stant temporal prior weights (constant prior), 'sia' with adaptive temporal prior weights (adaptive prior), 'sia' with global affine illumination model (global) and the proposed algorithm (proposed). Three error metrics used for evaluating performance of each case are the root mean square error (RMSE) of the relative pose error (RPE) where the step size is equal to one that measures the local accuracy of the given trajectory, RMSE of the absolute trajectory error (ATE) and the final position error divided by the whole length of the ground truth trajectory (% dt). RPE and ATE are proposed by [38] and broadly used for evaluating VO algorithms.

The experimental results are summarized in Table 3.2. It reports that the proposed algorithm attains the most accurate pose estimation result. In particular, the proposed method has decreased RMSE RPE by 43.5%, RMSE ATE by 54.6%, and % dt by 58.8% on average of other methods in Table 1. We observe that 'adaptive prior' and 'constant prior' show almost the same results as 'sia' case. This is because the frame rate (20 fps) relative to the motion is high enough to prevent the state vector from falling into a false minimum. Fig. 3 shows the $L_2$ norm of RPE and ATE throughout the flight. At the pair of images in Fig. 3.7, the proposed algorithm shows 0.02 m/s of RPE norm whereas 'sia' shows 0.0308 m/s and 'global' shows 0.0312 m/s that is seen at 105.6 seconds elapsed time in Fig. 3.7a.

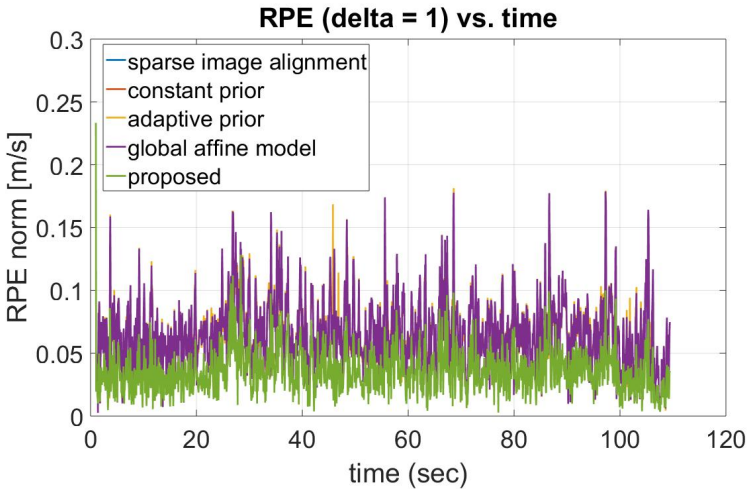
It is noteworthy to mention that the proposed method further reduces the pose estimation error by modeling local brightness changes the global model fails to account for. Also, the proposed method only requires 1.5-1.8 times computation time when compared to 'sia' case in MATLAB environment. As a result of the concatenation of relative poses, all five VO algorithms accumulates the ATE as in Fig. 3.7b. However, the proposed algorithm has reduced the accumulation by considering partial brightness changes.


(a) sample bucket pairs

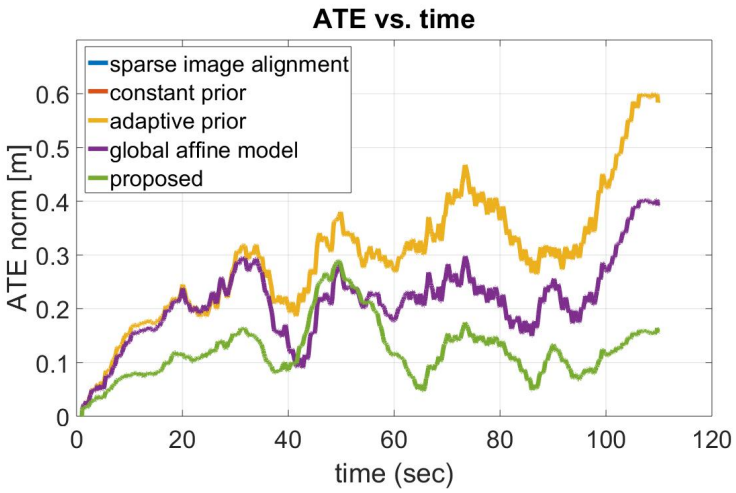



(b) intensity difference histogram

Figure 3.6: MH02 EuRoC dataset sample bucket pairs which are extracted at $t_1$ and $t_2$, and their intensity differences histogram




(a) $L_2$ norm of RPE versus flight time.




(b) $L_2$ norm of ATE versus flight time.

Figure 3.7: Pose estimation accuracy at EuRoC MH02 dataset, 'sia', 'constant prior' and 'adaptive prior' are almost identical.

Table 3.2: Performance comparison at the EuRoC MH02 dataset.

| | RMSE RPE [m/s] | RMSE ATE [m] | % dt [%] |
|---|---|---|---|
| sia | 0.0711 | 0.3213 | 0.93 |
| constant prior | 0.0710 | 0.3213 | 0.93 |
| adaptive prior | 0.0711 | 0.3212 | 0.93 |
| global | 0.0710 | 0.2256 | 0.62 |
| proposed | **0.0398** | **0.1350** | **0.25** |

### 3.3.3 Planetary rover datasets

We evaluate the proposed algorithm at the planetary rover dataset, ASRL dataset. It is recorded by sensors equipped on the rover at Devon island located at Canadian High Arctic which exhibits strong geological terrains with no artificial objects and structures. Due to its diverse geological terrains without vegetation, it is utilized for planetary exploration field tests [8]. The dataset provides grayscale images at 3 fps with ground truth initial attitude and synchronized differential global positional system (DPGS) positions.

Fig. 3.8, 3.9 shows sample images from the dataset featuring strong outdoor sunlight and large motion due to its low sampling time. More specifically, (a) and (b) in Fig. 3.8 are captured when the rover turned to the right, and only three-quarters of the previous image, Fig. 3.8a remains overlapped with the current image, Fig. 3.8b. Also, (a) and (b) in Fig. 3.6 exhibit partial illumination changes occurred by the projection of the outdoor sunlight into the lens even though Fig. 3.9(c,d) are temporally successive. Therefore, local illumination changes and motion priors should be considered in order to accurately estimate rover's pose. The parameter settings are the same as the MAV test except for bigger patch size ($5 \times 5$) and fewer buckets ($3 \times 3$). Also, note that we decide to employ Huber image weighting matrix [16] after trial and error to suppress large photometric errors. The ground truth and estimated trajectories are plotted in Fig. 5a and the rover drives 413-meter long trajectory for 11 minutes. We compare five algorithms as in the MAV dataset, and select error metrics as 3D position RMSE and % dt since the true attitude is not available in the dataset.

Table 3.3 summarizes the evaluation result that the proposed method outperforms the conventional methods; the proposed illumination model and the

adaptive prior weights have lowered the position RMSE by 79.5 % with regards to 'sia' case. Fig. 3.10 shows the evaluation results and several interesting observations. First, the large motion of the rover degrades the accuracy of pose estimation making high nonlinearity to the cost function, Eq. (3.7). Thus, 'sia' case shows the largest position error accumulation as shown in Fig. 3.10, hence the worst position accuracy, 58.6 m as summarized in Table. 3.3. The motion prior term in Eq. (3.8) holds the relative pose to stay near the previous motion stabilizing the estimator. Therefore, both 'constant prior' and 'adaptive prior' gives a more accurate estimation than 'sia' case. To reflect the importance of the prior term, we add the adaptive prior term to 'global' case to compare algorithms with the proposed one in fairness. Second, 'adaptive prior' shows 14.8 m better position accuracy than 'constant prior'. This is because we reflect the previous motion into the prior weighting matrix and the constant velocity residual is weighted accordingly. Third, even if the constant velocity model reduces the position error of the rover, both global and bucketed local illumination model reduce the error further. However, 'global' case cannot explain partial illumination changes such as image sequence in Fig. 3.9(a,b). We verify that the proposed algorithm is more robust to illumination changes than the global model through experimental evidence. The proposed algorithm yields the best position accuracy for the rover showing 4.2 m lower position RMSE than 'global' case. Lastly, a frame rate of a camera plays important role in DVO since the nonlinearity of the cost function is sensitive to how large overlapping region of a temporal image is.

(a)

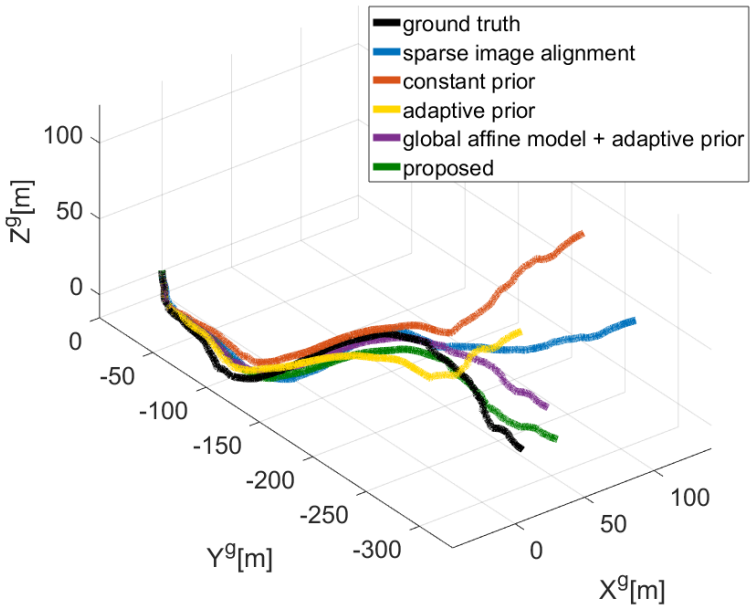
(b)

Figure 3.8: ASRL sample images, temporally consecutive image pairs that exhibit large motion, only three-quarters of the image is overlapped because of the large motion

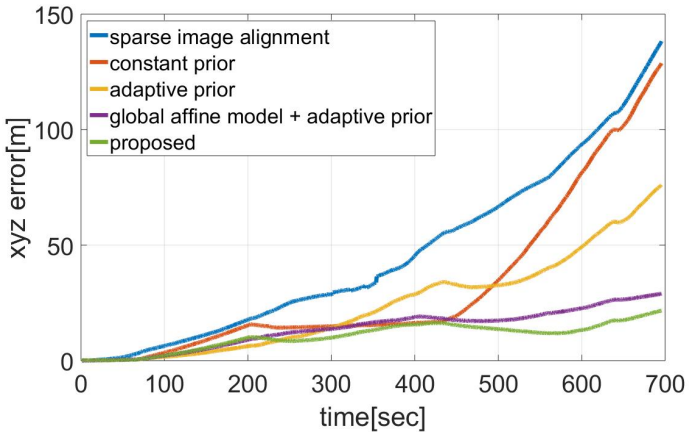
(a)

(b)

Figure 3.9: ASRL sample images, temporally consecutive image pairs that exhibit partial illumination change due to the sunlight




(a) Ground truth and estimated trajectory.




(b) xyz error versus time.

Figure 3.10: Pose estimation accuracy at ASRL dataset.

Table 3.3: Performance comparison at ASRL dataset.

| | Position RMSE [m] | % dt [%] |
|---|---|---|
| sia | 58.6 | 33.5 |
| constant prior | 46.3 | 31.2 |
| adaptive prior | 31.5 | 18.4 |
| global | 16.2 | 7.01 |
| proposed | **12.0** | **5.25** |

# Chapter 4

# Self-Calibrated Visual-Inertial Odometry

In this chapter, we present a visual-inertial odometry (VIO) with an online calibration using a stereo camera in planetary rover localization. We augment the state vector with extrinsic (rigid body transformation) and temporal (time-offset) parameters of a camera-IMU system in a framework of an extended Kalman filter. This is motivated by the fact that when fusing independent systems, it is practically crucial to obtain precise extrinsic and temporal parameters. Unlike the conventional calibration procedures, this method estimates both navigation and calibration states from naturally occurred visual point features during operation. We describe mathematical formulations of the proposed method, and it is evaluated through the author-collected dataset which is recorded by the commercially available visual-inertial sensor installed on the testing rover in the environment lack of vegetation and artificial objects. Our experimental results showed that 3D return position error as 1.54m of total 173m traveled and 10ms of time-offset with the online calibration, while 6.52m of return position error without the online calibration.

## 4.1 State representation

### 4.1.1 IMU state

The error state vector of the presented algorithm consists of the 15th order of IMU state ($\tilde{\mathbf{x}}_I$), the calibration states ($\tilde{\mathbf{x}}_C$) : cam-IMU time-offset, extrinsic parameter and the sliding window pose/velocity ($\tilde{\mathbf{x}}_S$) as in Eq. (4.1).

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\mathbf{x}}_I^{\mathrm{T}} & \tilde{\mathbf{x}}_C^{\mathrm{T}} & \tilde{\mathbf{x}}_S^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{4.1}$$

We define the error state as the difference between the true state ($\mathbf{x}$) and the estimated state ($\hat{\mathbf{x}}$) : $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$. In Eq. (4.1), the IMU-related filter state is given as follows,

$$\tilde{\mathbf{x}}_I = \begin{bmatrix} \tilde{\boldsymbol{\theta}}_{GB}^{\mathrm{T}} & {}^{G}\tilde{\mathbf{p}}_B^{\mathrm{T}} & {}^{G}\tilde{\mathbf{v}}_B^{\mathrm{T}} & \tilde{\mathbf{b}}_a^{\mathrm{T}} & \tilde{\mathbf{b}}_g^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{15} \tag{4.2}$$

$$\tilde{\mathbf{x}}_S = \begin{bmatrix} \tilde{\boldsymbol{\theta}}_{GB_i}^{\mathrm{T}} & {}^{G}\tilde{\mathbf{p}}_{B_i}^{\mathrm{T}} & {}^{G}\tilde{\mathbf{v}}_{B_i}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{9N} \tag{4.3}$$

As in Chapter 3, we denote the global frame as $\{G\}$, the camera frame as $\{C\}$, the body (IMU) frame as $\{B\}$, and the left superscript denotes the reference frame while the right subscript means the object frame. $\mathbf{p}$ and $\mathbf{v}$ mean the sensing platform's position and velocity, respectively. Also, $\mathbf{b}_a$ and $\mathbf{b}_a$ are biases for an accelerometer and gyroscope, respectively. The attitude error in Eq. (4.2) and (4.3) is defined as follow with the unit quaternion $\mathbf{q}$,

$$\mathbf{q}_{GB} = \begin{bmatrix} 1 \\ 1/2\tilde{\boldsymbol{\theta}}_{GB} \end{bmatrix} \otimes \hat{\mathbf{q}}_{GB} \tag{4.4}$$

where $\otimes$ is a quaternion multiplication.

In this paper, the sliding window is defined as a previous pose/velocity of the body frame when point features are captured. Eq. (4.3) is the sliding window-related filter states for i-th view where N is the total number of sliding windows.

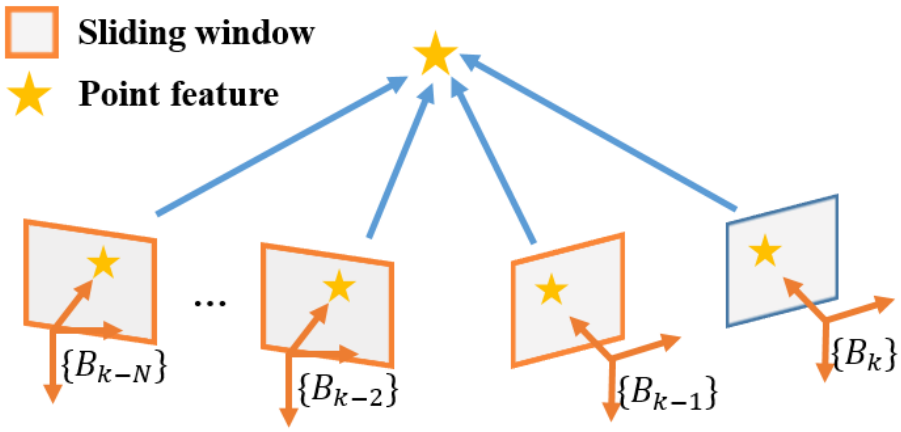



Figure 4.1: Graphical description of sliding windows

Fig. (4.1) illustrates this in detail where the blue-colored window indicates the current view ($\tilde{\mathbf{x}}_I$), while the orange windows are sliding windows ($\tilde{\mathbf{x}}_C$). Since we are interested in the localization problem, we do not include feature information in the filter state as suggested in [28].

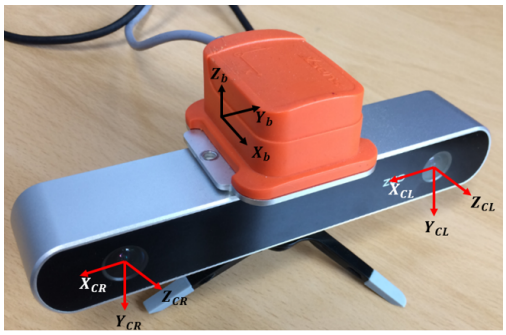
### 4.1.2 Calibration parameter state

The extrinsic spatial and temporal parameters of an IMU/Cam system are augmented in the filter state. Specifically, the calibration-related filter state is given as below

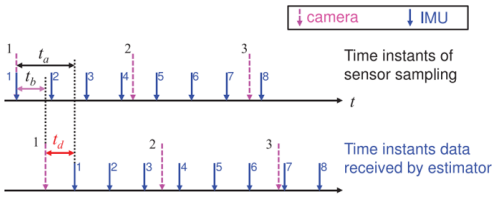
$$\tilde{\mathbf{x}}_C = \begin{bmatrix} \tilde{\boldsymbol{\theta}}_{CB}^{\mathrm{T}} & {}^{C}\tilde{\mathbf{p}}_B^{\mathrm{T}} & \tilde{t}_d \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^7 \tag{4.5}$$

where $\tilde{\boldsymbol{\theta}}_{CB}$ and ${}^{C}\tilde{\mathbf{p}}_B$ are the relative rotational and translational position error, respectively as shown in Fig. 4.2(a). $t_d$ in Eq. (4.5) is time-offset between an IMU and camera that is defined as

$$t_d \triangleq t_a - t_b \tag{4.6}$$




(a) Spatial extrinsic parameter




(b) Temporal extrinsic parameter [25]

Figure 4.2: Extrinsic parameter description in an IMU/Cam system

In the above expression, $t_a$ and $t_b$ stand for time-delay of an IMU and camera due to sensor's latency, respectively [25]. In the work of Li et al. [25], it is analytically proved that $t_d$ is observable (identifiable) referenced at the time-delay of an IMU.

## 4.2 State-propagation

The IMU measurements are modeled as Eq. (4.7), (4.8) with the zero-mean white Gaussian noise process ($\mathbf{n}$), and the random walk process ($\mathbf{b}$).

$$\mathbf{a}_m(t) = {}^G\mathbf{R}_B(t)({}^G\mathbf{a}(t) - {}^G\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t) \tag{4.7}$$

$$\mathbf{w}_m(t) = \mathbf{w}_t(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \tag{4.8}$$

Where ${}^G\mathbf{a}(t)$ is the true acceleration of the sensing platform, ${}^G\mathbf{g}$ is the global gravity that is approximately $\begin{bmatrix}0 & 0 & 9.81\end{bmatrix}^{\mathrm{T}}$ $m/s^2$ in the gravity-aligned reference frame. According to the sensor modeling, the continuous-time system dynamics is given as

$$\dot{\mathbf{q}}_{GB}(t) = \frac{1}{2}\mathbf{q}_{GB}(t) \otimes (\mathbf{w}_m - \mathbf{b}_g(t)) \tag{4.9}$$

$$ {}^{G}\dot{\mathbf{p}}_B(t) = {}^{G}\mathbf{v}_B(t) \tag{4.10} $$

$$ {}^{G}\dot{\mathbf{v}}_B(t) = {}^{G}\mathbf{R}_B(t)\left(\mathbf{a}_m - \mathbf{b}_a(t)\right) + {}^{G}\mathbf{g} \tag{4.11} $$

$$ \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) \tag{4.12} $$

$$ \dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t) \tag{4.13} $$

In Eq. (4.12), (4.13), $\mathbf{n}_{wa}$ and $\mathbf{n}_{wg}$ are zero-mean white Gaussian noise processes. The nominal states are computed by a numerical integration based on Eq. (4.9) - (4.11).

The error-state system model in continuous time is as follow, while calibration-related states are assumed to be constant over time.

$$ \dot{\tilde{\mathbf{x}}}_I(t) = \mathbf{F}_I(t)\tilde{\mathbf{x}}_I(t) + \mathbf{G}_I(t)\mathbf{n}_I(t) \tag{4.14} $$

$$ \mathbf{n}_I(t) = \begin{bmatrix} \mathbf{n}_a^{\mathrm{T}} & \mathbf{n}_g^{\mathrm{T}} & \mathbf{n}_{wa}^{\mathrm{T}} & \mathbf{n}_{wg}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{4.15} $$

The Jacobian matrices in Eq. (4.14) can be derived using Taylor series expansion up to the 1st order.

$$ \mathbf{F}_I(t) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -{}^{G}\hat{\mathbf{R}}_B(t) \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ -\left[{}^{G}\hat{\mathbf{R}}_B(t)(\mathbf{a}_m - \hat{\mathbf{b}}_a(t))\right]_{\times} & \mathbf{0} & \mathbf{0} & -{}^{G}\hat{\mathbf{R}}_B(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4.16} $$

$$ \mathbf{G}_I(t) = \begin{bmatrix} \mathbf{0} & -{}^{G}\hat{\mathbf{R}}_B(t) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -{}^{G}\hat{\mathbf{R}}_B(t) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \tag{4.17} $$

where $[\cdot]_\times$ is a skew-symmetric matrix operator as in Chapter 3, and $\mathbf{I}_3$ stands for a 3 by 3 identity matrix. The covariance matrix ($\mathbf{P}_I$) of the system is propagated through matrices in discrete time domain.

$$\mathbf{P}_{I_{k+1}} = \mathbf{\Phi}_{I_k} \mathbf{P}_{I_k} \mathbf{\Phi}_{I_k}^{\mathrm{T}} + \mathbf{Q}_k \tag{4.18}$$

The state-transition matrix in discrete time given the sampling time is

$$\mathbf{\Phi}_{I_k} = \mathbf{\Phi}(t_{k+1}, t_k) \tag{4.19}$$

such that

$$\dot{\mathbf{\Phi}}(\tau, t_k) = \mathbf{F}_I(\tau)\mathbf{\Phi}(\tau, t_k), \quad \tau \in [t_k, t_{k+1}] \tag{4.20}$$

Also, $\mathbf{Q}_k$ is computed as follow,

$$\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1}, \tau)\mathbf{G}_I(\tau)\mathbf{Q}\mathbf{G}_I^{\mathrm{T}}(\tau)\mathbf{\Phi}^{\mathrm{T}}(t_{k+1}, \tau)d\tau \tag{4.21}$$

where $\mathbf{Q}$ is a power spectral density matrix which is obtained from the specification of an IMU such that

$$\mathbb{E}\left[\mathbf{n}_I(t)\mathbf{n}_I^{\mathrm{T}}(\tau)\right] = \mathbf{Q}\,\delta(t-\tau) \tag{4.22}$$

where $\delta$ is Dirac delta function.

The covariance matrix of the whole state is partitioned as

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{I_k} & \mathbf{P}_{IA_k} \\ \mathbf{P}_{AI_k} & \mathbf{P}_{A_k} \end{bmatrix} \tag{4.23}$$

where the subscript $A$ includes filter states other than IMU-related states. Then, the covariance matrix is propagated by,

$$\mathbf{P}_{k+1} = \begin{bmatrix} \mathbf{P}_{I_{k+1}} & \mathbf{\Phi}_{I_k}\mathbf{P}_{IA_k} \\ \mathbf{P}_{AI_k}\mathbf{\Phi}_{I_k}^{\mathrm{T}} & \mathbf{P}_{A_k} \end{bmatrix} \tag{4.24}$$

## 4.3 Measurement-update

### 4.3.1 Point feature measurement

To deal with the calibration parameters of a IMU/Cam system, their error state should be modeled in the measurement model. Note that the extrinsic parameter is observable under the point features [11], and the time-offset is also observable up to the time referenced at an IMU [25]. These motivate us to jointly estimate the parameters along with the navigation solution in a stereo vision scenario which provides reliable depth information.

In order to build constraints among multiple views for point features, sliding window poses/velocity should be augmented to the state vector. Propagating the current IMU state up to the time-offset ($t_d$), the sliding window state is as follow,

$$\tilde{\mathbf{x}}_{S_i} = \begin{bmatrix} \tilde{\boldsymbol{\theta}}_{GB_i}^{\mathrm{T}}(t_n) & {}^{G}\tilde{\mathbf{p}}_{B_i}^{\mathrm{T}}(t_n) & {}^{G}\tilde{\mathbf{v}}_{B_i}^{\mathrm{T}}(t_n) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{9N} \tag{4.25}$$

To simplify notations, we define the actual timestamp when the image is captured as $t_n \triangleq t + t_d$ where $t$ is the nominal image timestamp. Accordingly, the Jacobian matrix with regard to the IMU state is

$$\tilde{\mathbf{x}}_{S_i} \cong \begin{bmatrix} \mathbf{I}_9 & \mathbf{0}_{9\times 6} & \mathbf{0}_{9\times 6} & \mathbf{J}_{t_d} & \mathbf{0}_9 \end{bmatrix} \tilde{\mathbf{x}}_I(t_n) \tag{4.26}$$

Where $\mathbf{J}_{t_d}$ is the Jacobian matrix related to the time-offset, and can be derived using 1st order approximation,

$$\mathbf{J}_{t_d} = \begin{bmatrix} {}^{G}\hat{\mathbf{R}}_B(\hat{t}_n)(\mathbf{w}_m - \hat{\mathbf{b}}_g(\hat{t}_n)) \\ {}^{G}\hat{\mathbf{v}}_B(\hat{t}_n) \\ {}^{G}\hat{\mathbf{R}}_B(\hat{t}_n)(\mathbf{a}_m - \hat{\mathbf{b}}_a(\hat{t}_n)) + {}^{G}\mathbf{g} \end{bmatrix} \in \mathbb{R}^9 \tag{4.27}$$

Assuming that a stereo camera is well calibrated in advance, the point fea-

ture measurement model is

$$\mathbf{z}(t) = \begin{bmatrix} 1/Z_L\mathbf{I}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & 1/Z_R\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} {}^{C_L}\mathbf{p}_f[1:2] \\ {}^{C_R}\mathbf{p}_f[1:2] \end{bmatrix} + \mathbf{n}_z \tag{4.28}$$

$${}^{C_L}\mathbf{p}_f(t_n) = \begin{bmatrix} X_L & Y_L & Z_L \end{bmatrix}^{\mathrm{T}} \tag{4.29}$$

$${}^{C_R}\mathbf{p}_f(t_n) = \begin{bmatrix} X_R & Y_R & Z_R \end{bmatrix}^{\mathrm{T}} \tag{4.30}$$

with zero-mean white Gaussian noise process, $\mathbf{n}_z$. In this expression, $\{C_L\}$ and $\{C_R\}$ denote the left/right camera coordinate frame. Note that in Eq. (4.29), (4.30), the instance of time is at $t_n$ for the feature position.

### 4.3.2 Measurement error modeling

The linearized measurement model with a single point feature is given by,

$$\mathbf{r} = \mathbf{z}(t) - \hat{\mathbf{z}}(\hat{t}_n) \cong \mathbf{H}(\hat{t}_n)\tilde{\mathbf{x}}(\hat{t}_n) + \mathbf{n}_z \tag{4.31}$$

where $\mathbf{H}$ matrix is the measurement Jacobian matrix. Specifically, this matrix is computed from the left and right measurements,

$$\mathbf{H}(\hat{t}_n) = \left( \frac{\partial \mathbf{z}}{\partial {}^{C_L}\mathbf{p}_f} \frac{\partial {}^{C_L}\mathbf{p}_f}{\partial \mathbf{x}} + \frac{\partial \mathbf{z}}{\partial {}^{C_R}\mathbf{p}_f} \frac{\partial {}^{C_R}\mathbf{p}_f}{\partial \mathbf{x}} \right)_{\hat{t}_n} \tag{4.32}$$

The derivatives with respect to the feature point are easily derived as,

$$\frac{\partial \mathbf{z}}{\partial {}^{C_L}\mathbf{p}_f} = \begin{bmatrix} \frac{1}{Z_L} & 0 & -\frac{X_L}{Z_L^2} \\ 0 & \frac{1}{Z_L} & -\frac{Y_L}{Z_L^2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.33}$$

$$\frac{\partial \mathbf{z}}{\partial {}^{C_R}\mathbf{p}_f} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{Z_R} & 0 & -\frac{X_L}{Z_R^2} \\ 0 & \frac{1}{Z_R} & -\frac{Y_L}{Z_R^2} \end{bmatrix} \tag{4.34}$$

To derive derivatives of feature positions with respect to the filter state in Eq. (4.32), the global pose is perturbed up to the 1st order Taylor series expansion with respect to the time-offset,

$${}^{G}\mathbf{R}_B(t_n) \cong {}^{G}\mathbf{R}_B(\hat{t}_n) + {}^{G}\dot{\mathbf{R}}_B(\hat{t}_n)\tilde{t_d} \tag{4.35}$$

Perturbing the above equation with respect to the current attitude estimates yields,

$$\begin{aligned} {}^{G}\mathbf{R}_B(t_n) \cong \left(\mathbf{I}_3 + \left[\tilde{\boldsymbol{\theta}}_{GB}(\hat{t}_n)\right]_\times\right) {}^{G}\mathbf{R}_B(\hat{t}_n) \\ + {}^{G}\mathbf{R}_B(\hat{t}_n)\left[\mathbf{w}_m - \tilde{\mathbf{b}}_g(\hat{t}_n)\right]_\times \tilde{t}_d \end{aligned} \tag{4.36}$$

Likewise,

$${}^{G}\mathbf{p}_B(t_n) \cong {}^{G}\mathbf{p}_B(\hat{t}_n) + {}^{G}\mathbf{v}_B(\hat{t}_n)\tilde{t_d} \tag{4.37}$$

$${}^{G}\mathbf{p}_B(t_n) \cong {}^{G}\hat{\mathbf{p}}_B(\hat{t}_n) + {}^{G}\tilde{\mathbf{p}}_B(\hat{t}_n) + {}^{G}\hat{\mathbf{v}}_B(\hat{t}_n)\tilde{t_d} \tag{4.38}$$

The linearization in Eq. (4.36), (4.38) enables us to model the time-offset in the measurement model.

The feature position referenced at the left camera frame can be expressed as follow,

$${}^{C_L}\mathbf{p}_f(t_n) = {}^{C_L}\mathbf{R}_B\ {}^{B}\mathbf{R}_G(t_n)\left({}^{G}\mathbf{p}_f - {}^{G}\mathbf{p}_B(t_n)\right) + {}^{C_L}\mathbf{p}_B \tag{4.39}$$

Substituting Eq. (4.36), (4.38) into Eq. (4.39) yields,

$$
\begin{aligned}
{}^{C_L}\tilde{\mathbf{p}}_f(t_n) \cong\ & {}^{C_L}\hat{\mathbf{R}}_G(\hat{t}_n)\ {}^{G}\tilde{\mathbf{p}}_f + {}^{C_L}\tilde{\mathbf{p}}_B - \left[{}^{C_L}\tilde{\mathbf{R}}_f(\hat{t}_n)({}^{G}\hat{\mathbf{p}}_f - {}^{G}\hat{\mathbf{p}}_B(\hat{t}_n))\right]_\times \tilde{\boldsymbol{\theta}}_{C_L B} \\
-{}^{C_L}\hat{\mathbf{R}}_B & \left(\left[w_m - \hat{\mathbf{b}}_g(\hat{t}_n)\right]_\times {}^{B}\hat{\mathbf{R}}_G(\hat{t}_n)({}^{G}\hat{\mathbf{p}}_f - {}^{G}\hat{\mathbf{p}}_B(\hat{t}_n)) + {}^{B}\hat{\mathbf{R}}_G(\hat{t}_n)\ {}^{B}\hat{\mathbf{v}}_G(\hat{t}_n)\right)\ \tilde{t}_d \\
& + {}^{C_L}\hat{\mathbf{R}}_G(\hat{t}_n)\left[{}^{G}\hat{\mathbf{p}}_f - {}^{G}\hat{\mathbf{p}}_B(\hat{t}_n)\right]_\times \tilde{\boldsymbol{\theta}}_{GB}(\hat{t}_n) - {}^{G}\hat{\mathbf{R}}_B(\hat{t}_n){}^{G}\tilde{\mathbf{p}}_B(\hat{t}_n)
\end{aligned}
\tag{4.40}
$$

Therefore, $\partial\ {}^{C_L}\mathbf{p}_f/\partial\mathbf{x}$ can be obtained with the corresponding filter state in Eq. (4.40). Since we assume that ${}^{C_L}\mathbf{T}_{C_R}$ is known, the derivatives of the feature position referenced at the right camera frame is compuated as follow,

$$
\frac{\partial^{C_R}\mathbf{p}_f}{\partial\mathbf{x}} = {}^{C_R}\mathbf{R}_{C_L}\frac{\partial^{C_L}\mathbf{p}_f}{\partial\mathbf{x}} \tag{4.41}
$$

Using Jacobian matrices obtained in Eq. (4.40), the linearized model in Eq. (4.31) is written as,

$$
\mathbf{r} \cong\ \mathbf{H}_S(\hat{t}_n)\tilde{\mathbf{x}}_S(\hat{t}_n) + \mathbf{H}_C(\hat{t}_n)\tilde{\mathbf{x}}_C + \mathbf{H}_f(\hat{t}_n)\tilde{\mathbf{x}}_f + \mathbf{n}_z \tag{4.42}
$$

where $\mathbf{H}$ matrices are the corresponding Jacobian matrices, i.e.,

$$
\mathbf{H}_S(\hat{t}_n) = \left(\frac{\partial\mathbf{z}}{\partial^{C_L}\mathbf{p}_f}\frac{\partial^{C_L}\mathbf{p}_f}{\partial\mathbf{x}_S} + \frac{\partial\mathbf{z}}{\partial^{C_R}\mathbf{p}_f}\frac{\partial^{C_R}\mathbf{p}_f}{\partial\mathbf{x}_S}\right)_{\hat{t}_n} \tag{4.43}
$$

$$
\mathbf{H}_C(\hat{t}_n) = \left(\frac{\partial\mathbf{z}}{\partial^{C_L}\mathbf{p}_f}\frac{\partial^{C_L}\mathbf{p}_f}{\partial\mathbf{x}_C} + \frac{\partial\mathbf{z}}{\partial^{C_R}\mathbf{p}_f}\frac{\partial^{C_R}\mathbf{p}_f}{\partial\mathbf{x}_C}\right)_{\hat{t}_n} \tag{4.44}
$$

$$
\mathbf{H}_f(\hat{t}_n) = \left(\frac{\partial\mathbf{z}}{\partial^{C_L}\mathbf{p}_f}\frac{\partial^{C_L}\mathbf{p}_f}{\partial\mathbf{x}_f} + \frac{\partial\mathbf{z}}{\partial^{C_R}\mathbf{p}_f}\frac{\partial^{C_R}\mathbf{p}_f}{\partial\mathbf{x}_f}\right)_{\hat{t}_n} \tag{4.45}
$$

To be specific, Jacobian matrices in Eq. (4.43) - (4.45) can be obtained from Eq. (4.40),

$$
\frac{\partial^{C_L}\mathbf{p}_f}{\partial\mathbf{x}_S} = \begin{bmatrix}{}^{C_L}\hat{\mathbf{R}}_G(\hat{t}_n)\left[{}^{G}\hat{\mathbf{p}}_f - {}^{G}\hat{\mathbf{p}}_B(\hat{t}_n)\right]_\times & -{}^{G}\hat{\mathbf{R}}_B(\hat{t}_n) & \mathbf{0}_3\end{bmatrix} \tag{4.46}
$$

$$\frac{\partial^{C_L}\mathbf{p}_f}{\partial \mathbf{x}_C} = \begin{bmatrix} -\left[{}^{C_L}\tilde{\mathbf{R}}_f(\hat{t}_n)({}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_B(\hat{t}_n))\right]_\times & \mathbf{I}_3 & \mathbf{M} \end{bmatrix} \tag{4.47}$$

$$\mathbf{M} = -{}^{C_L}\hat{\mathbf{R}}_B \left( \left[ w_m - \hat{\mathbf{b}}_g(\hat{t}_n) \right]_\times {}^B\hat{\mathbf{R}}_G(\hat{t}_n)({}^G\hat{\mathbf{p}}_f - {}^G\hat{\mathbf{p}}_B(\hat{t}_n)) + {}^B\hat{\mathbf{R}}_G(\hat{t}_n)\ {}^B\hat{\mathbf{v}}_G(\hat{t}_n) \right) \tag{4.48}$$

$$\frac{\partial^{C_L}\mathbf{p}_f}{\partial \mathbf{x}_f} = {}^{C_L}\hat{\mathbf{R}}_G(\hat{t}_n) \tag{4.49}$$

To eliminate feature information in the measurement equation, Eq. (4.42) is projected into the left nullspace of the feature-related Jacobian matrix ($\mathbf{H}_f$) [28].

$$\mathbf{A}^{\mathrm{T}}\mathbf{r} \cong \mathbf{A}^{\mathrm{T}}\mathbf{H}_S(\hat{t}_n)\tilde{\mathbf{x}}_S(\hat{t}_n) + \mathbf{A}^{\mathrm{T}}\mathbf{H}_C(\hat{t}_n)\tilde{\mathbf{x}}_C(\hat{t}_n) + \mathbf{A}^{\mathrm{T}}\mathbf{n}_z \tag{4.50}$$

In this expression, $\mathbf{A}^{\mathrm{T}}$ is the left nullspace matrix of $\mathbf{H}_f$ such that

$$\mathbf{A}^{\mathrm{T}}\mathbf{H}_f = \mathbf{0} \tag{4.51}$$

Note that since $\mathbf{H}_f \in \mathbb{R}^{4N\times 3}$, the left nullspace exists in general with the dimension of $\mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{(4N-3)\times 4N}$ where $N$ is the number of sliding window as mentioned before. We can rewrite Eq. (4.50) omitting the time instances for simplicity as

$$\mathbf{r}_o = \mathbf{H}_{S_o}\tilde{\mathbf{x}}_S + \mathbf{H}_{C_o}\tilde{\mathbf{x}}_C + \mathbf{n}_o \tag{4.52}$$

Therefore,

$$\mathbf{R}_o = \mathbf{H}_u\tilde{\mathbf{x}} + \mathbf{n}_o \tag{4.53}$$

$$\mathbf{H}_u = \begin{bmatrix} \mathbf{0} & \mathbf{H}_{S_o} & \mathbf{H}_{C_o} \end{bmatrix} \tag{4.54}$$

The Kalman gain is computed as

$$\mathbf{K} = \mathbf{P}\mathbf{H}_u^{\mathrm{T}} \left(\mathbf{H}_u \mathbf{P} \mathbf{H}_u^{\mathrm{T}} + \mathbf{R}_o\right)^{-1} \tag{4.55}$$

$$\mathbf{R}_o = \mathbb{E}(\mathbf{n}_o \mathbf{n}_o^{\mathrm{T}}) \tag{4.56}$$

Then, the state correction is obtained as,

$$\Delta\mathbf{x} = \mathbf{K}\mathbf{r}_o \tag{4.57}$$

The filter state is corrected as below,

$$\mathbf{x}^+ = \mathbf{x} \oplus \boldsymbol{\Delta}\mathbf{x} \tag{4.58}$$

For each of the filter state,

$$\mathbf{x}_I^+ = \begin{bmatrix} \Delta\mathbf{q} \otimes \mathbf{q}_{GB} \\ {}^G\mathbf{p}_B + \Delta\mathbf{p} \\ {}^G\mathbf{v}_B + \Delta\mathbf{v} \\ \mathbf{b}_a + \Delta\mathbf{b}_a \\ \mathbf{b}_g + \Delta\mathbf{b}_g \end{bmatrix} \tag{4.59}$$

$$\mathbf{x}_C^+ = \begin{bmatrix} \Delta\mathbf{q}_{CB} \otimes \mathbf{q}_{CB} \\ {}^C\mathbf{p}_B + \Delta\mathbf{p}_{CB} \\ t_d + \Delta t_d \end{bmatrix} \tag{4.60}$$

$$\mathbf{x}_S^+ = \begin{bmatrix} \Delta\mathbf{q}_i \otimes \mathbf{q}_{GB} \\ {}^G\mathbf{p}_{B_i} + \Delta\mathbf{p}_i \\ {}^G\mathbf{v}_{B_i} + \Delta\mathbf{v}_i \end{bmatrix} \tag{4.61}$$

The covariance matrix of the system is updated as follow,

$$\mathbf{P}^+ = (\mathbf{I} - \mathbf{K}\mathbf{H}_u)\mathbf{P}(\mathbf{I} - \mathbf{K}\mathbf{H}_u)^{\mathrm{T}} + \mathbf{K}\mathbf{R}_o\mathbf{K}^{\mathrm{T}} \tag{4.62}$$

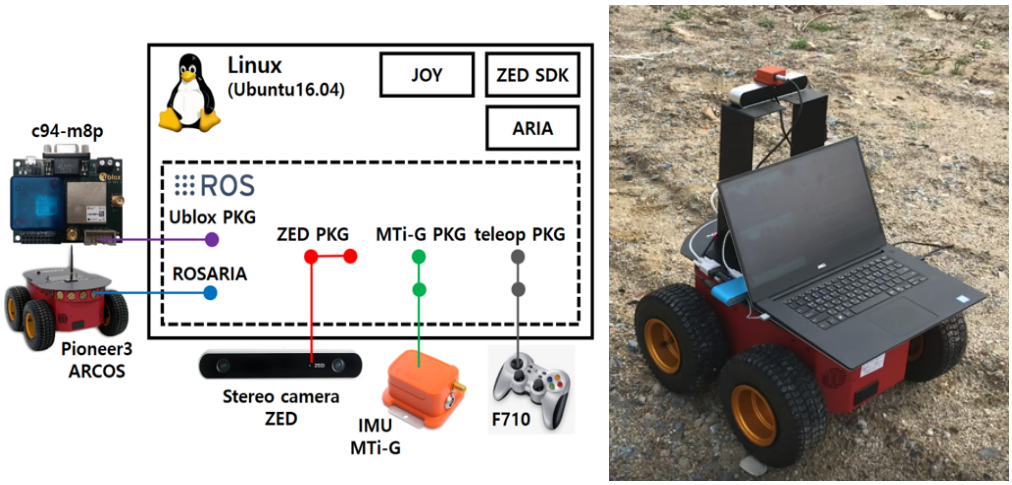
This concludes the filter update procedure.

## 4.4 Experimental results

### 4.4.1 Hardware setup

The testing rover in Fig. 4.3(b) consists of Pioneer3-AT (rover platform), Xsens MTi-300 (IMU), ZED stereo camera, and the onboard computer for the purpose of data recording. While IMU outputs its data at 200Hz, the stereo camera gives 1280x720 grey images at 15Hz. The visual-inertial sensor suite is shown in Fig. 4.2(b). As shown in Fig. 4.3(a), all sensors are integrated into ROS (Robot Operating System) which is robotics middleware. Although both sensors are timestamped under the ROS environment, the nature of the separate system motivates us to estimate the spatial and temporal extrinsic parameters. The initial guess of the extrinsic parameter was computed using Kalibr toolbox [9]. Also, a human pilot drove the testing rover returning to the starting point to quantify return position error. The typical environment of the site is shown in Fig. 4.4 which lacks artificial object and vegetation.

The rover platform (Pioneer 3-AT) in Fig. 4.5 is a four-wheel-drive rover, and it provides space for mounting sensors or equipments on its upper deck. Each motor is equipped with an optical encoder that measures the angular displacement of the motor using two signals with a phase difference of 90 degrees so that the speed of the rover can be measured. In addition, a user control panel is provided on the upper deck to confirm the current state of the rover and the connection state with the navigation computer. The rover has a microcontroller (MCU) as an onboard processor. Low-level tasks such as motor drive and measurement data processing are performed by the MCU's firmware, Advanced Robot Control and Operations Software (ARCOS), while high-level tasks such as command transfer from the user to the rover are performed in the navigation computer.




(a) Testing rover system architecture

(b) Testing rover

Figure 4.3: Rover field testing hardware setup

Figure 4.4: A typical image in the testing area

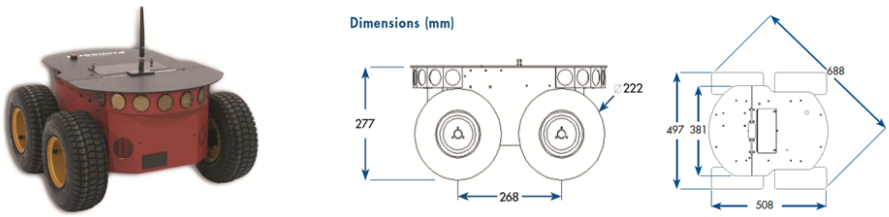



Figure 4.5: Rover platform : Pioneer 3-AT

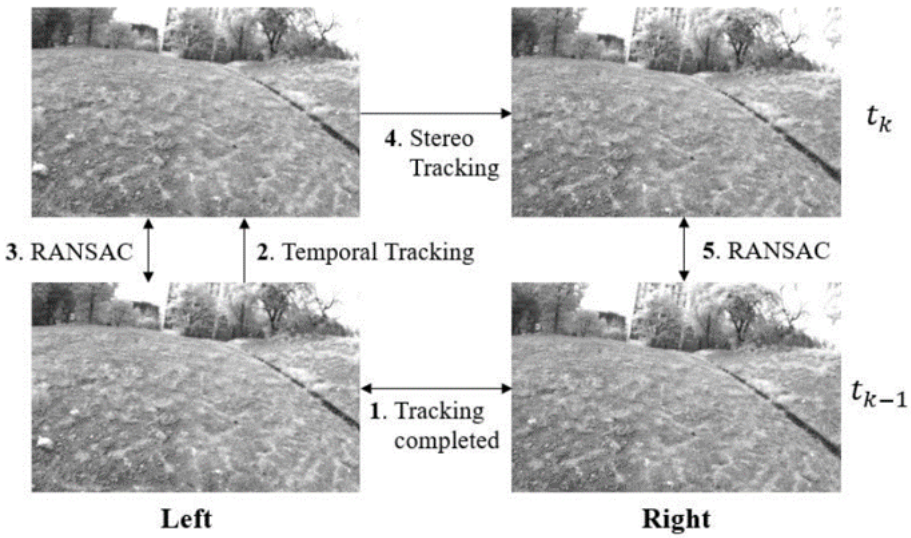



Figure 4.6: Feature tracking strategy using stereo images

### 4.4.2 Vision front-end design

We describe details of the vision front-end implementation in this section. Features are provided to the estimator when either tracking fails or the number of tracks exceeds a user-defined maximum sliding window. To obtain reliable sets of feature tracks, we design a stereo feature tracker shown in Fig. 4.6 as similar to [39]. Assuming that the feature correspondence at $t_{k-1}$ is obtained, features on the left image are tracked to the next time step $t_k$, then 8-point RANSAC eliminates outlier sets. Survived inliers on the left image are kept tracked to the right image. Again, 8-point RANSAC detects outliers between temporal right images at $t_{k-1}$ and $t_k$ . In the only case when the feature is successfully tracked both the temporal and static tracking, the feature is fed to the estimator. In contrast to a monocular case, the stereo features give scale information due to the baseline. Specifically, we triangulate feature points from the farthest two-view; for instance, the oldest frame in the left and the latest frame in the right before the multi-view triangulation. This strategy enables us to compute the feature depth, even the sensors are in static.

### 4.4.3 Rover field testing

To test the presented VIO, the testing rover traveled the total distance of 173m for 224 seconds commanded by the human pilot. The testing site mainly consisted of soil with small rocks where typical images are shown in Fig .4.4. To compute an initial attitude with respect to the navigation frame (NED-frame), outputs of the accelerometer at the first 2 seconds were used. Also, the testing rover started from the static state; the initial velocity was set to zero.

To quantify the performance of the algorithm, we compare return position errors among three cases: "full calibration ($t_d$ + ${}^{C}\mathbf{T}_{B}$)", "partial calibration

(only $t_d$)" and "no calibration". Table. 4.1 shows 3D return position of 3 cases in the Cartesian coordinate in which the starting point was $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^{\mathrm{T}}$ m. As expected the full calibration yields the best performance (2-norm) that is 76.4% error decrease when compared to the no calibration. It is interesting to note that the z-axis position of the partial calibration drifted up to -4.07m. We argue that this is due to the inaccurate extrinsic parameter that is computed beforehand. Also, Fig. 4.7 plots the entire estimated 2D trajectories of all cases. It is clearly seen in Fig. 4.7 that the no calibration largely drifts after the first 180 deg turning when compared the others.

Fig. 4.8 plots the estimated time-offset with its 3-sigma envelopes in the full calibration scenario. After quick convergence at the beginning, it converges to -10.3ms. Even though we do not have the true value of the time-offset, it is clearly seen that the uncertainty converges as time goes; that is a consistent result with the observability analysis of Li et al. [25]. Remind that the sampling time of images is 66.7ms (15Hz), thus the time-offset is not negligible.

Fig. 4.9 and 4.10 show 3 standard deviations for the IMU-Cam extrinsic parameter for each axis in the full calibration. We set the initial standard deviation as 1deg and 5mm respectively to cover the calibration uncertainty. Although we do not have the true value, the standard deviations are decreased as the filter is updated that is consistent results to the observability analysis in [11].

Table 4.1: 3D return position error for 3 cases

| | No Calibration | Partial Calibration | Full Calibration |
|---|---|---|---|
| xyz Return position [m] | -0.8229;<br>3.3352;<br>-5.5402 | 0.3627;<br>0.1956;<br>-4.0725 | -1.1922;<br>0.5711;<br>-0.7856 |
| 2-norm [m] | 6.52 | 4.09 | **1.54** |

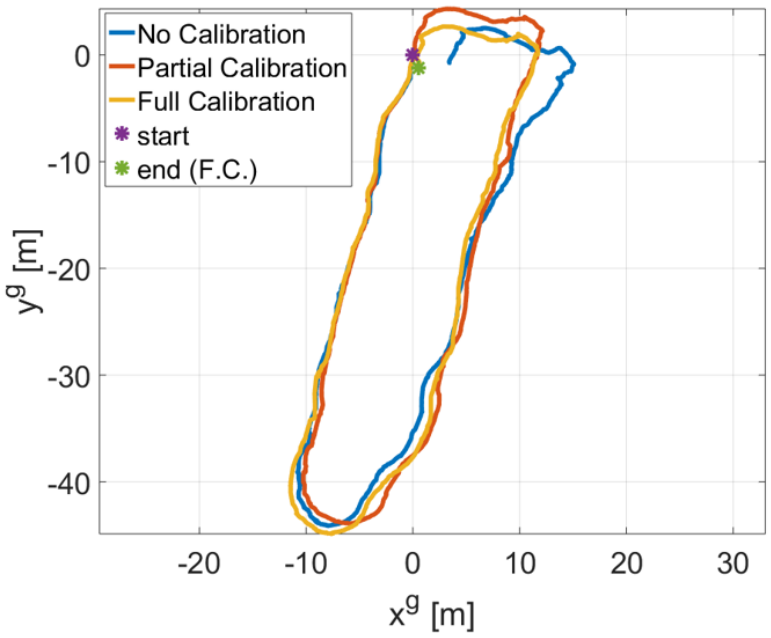



Figure 4.7: Estimated 2D trajectory with the online calibration

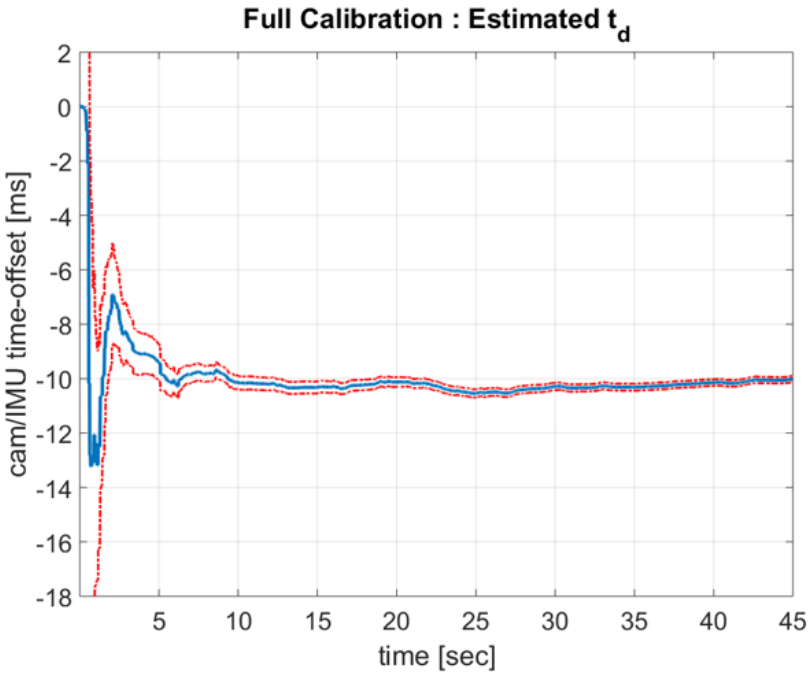



Figure 4.8: Estimated IMU-Cam time-offset with 3-sigma envelope

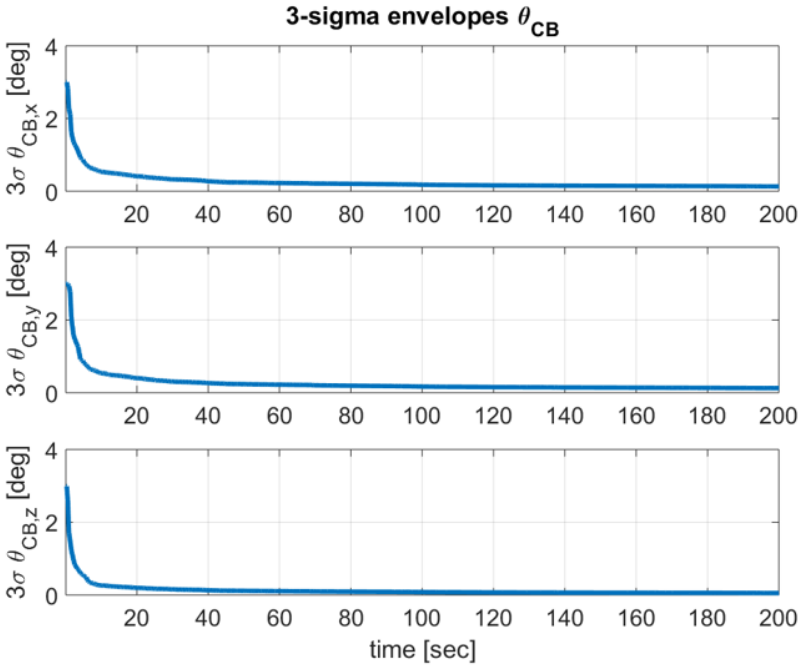



Figure 4.9: IMU-Cam relative attitude 3-sigma envelopes

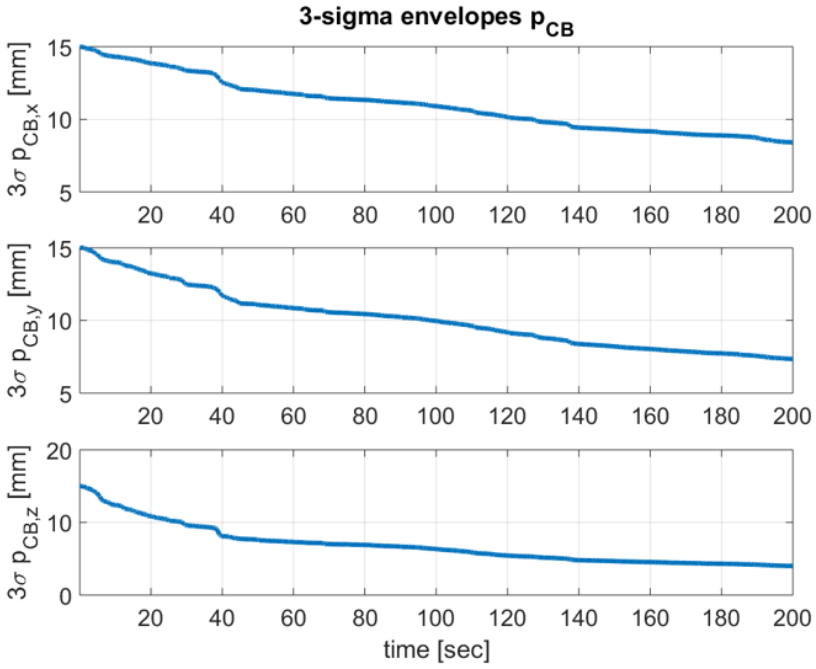



Figure 4.10: IMU-Cam relative position 3-sigma envelopes

# Chapter 5

# Conclusions

## 5.1 Conclusion and summary

In this paper, we have posed the problems about a localization task of a rover using an IMU and cameras. First, the Lambertian surface assumption is violated due to brightness changes of an image in a framework of the direct visual odometry. Second, the extrinsic calibration parameters would cause a big challenge when fusing measurements from an IMU and camera.

To improve the localization performance of a rover, we have proposed the bucketed local illumination model and the adaptive prior weight in patch-based DVO framework. The proposed illumination model does not require depths for all pixels in the image while accounting for both global and local illumination changes. A further advantage of the proposed model is that it does not exploit artificial planar patches but small patches that are assumed to be planar patches in smooth surfaces. Furthermore, the adaptive prior weight reflects the fact that a fast-moving-object gives more confidence to the constant velocity model than a slow-moving-object statistically. Finally, we have evaluated our algorithm in the MAV and the planetary rover dataset where the camera's automatic exposure and the strong outdoor sunlight induce partial and sudden illumination changes. Our experimental result reports that the proposed algorithm is robust to illumination changes and large motions showing much better

pose/position accuracy than the global illumination model.

In the case of the calibration issue in a visual-inertial system, we have presented the online calibration stereo VIO using naturally occurring point features in which the state vector is augmented by the time-offset and extrinsic parameters. To evaluate the presented VIO, the testing rover with the commercially available visual-inertial sensor recorded the dataset. Our experimental results have shown that when fusing independent sensors their extrinsic calibration is important; the online calibration method reduced the rover's return position error by 76.4% with respect to the no calibration method. Moreover, we experimentally showed that the time-offset and extrinsic parameter were observable under point features that is consistent with the observability analysis.

## 5.2 Future works

The proposed localization method can be further improved on the two aspects.

- **Photometric measurement model**

  The proposed bucketed illumination model is tested through a framework of direct visual odometry. However, it can be combined with a pipeline of the self-calibrated VIO. It is known that the photometric error model conveys more information than the reprojection model, but the former one is sensitive to illumination changes. It is expected that our proposed model would yield more robust localization results in an outdoor environment.

- **Initialization**

  In this study, we assume that reasonable initial conditions of the filter state are accessible. To be specific, we begin our algorithm from a static condition. However, in a real-world setting, this assumption can

be violated, or even worse, the filter would be failed due to a rapid motion or a huge occlusion in images. This motivates a robust initialization procedure— initial biases of an IMU, velocity, and gravity direction should be recovered using a constraint formulated by a sequence of images. This enables that the algorithm is bootstrapped even under a non-static condition by initializing its state vector.

# Bibliography

[1] Raymond E Arvidson, Karl D Iagnemma, Mark Maimone, Abigail A Fraeman, Feng Zhou, Matthew C Heverly, Paolo Bellutta, David Rubin, Nathan T Stein, John P Grotzinger, et al. Mars science laboratory curiosity rover megaripple crossings up to sol 710 in gale crater. *Journal of Field Robotics*, 34(3):495–518, 2017.

[2] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[3] Yang Cheng, Mark W Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. *IEEE Robotics and Automation magazine*, 13 (2):54, 2006.

[4] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1456, 2013.

[5] Jakob Engel, Jörg Stückler, and Daniel Cremers. Large-scale direct slam with stereo cameras. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1935–1942. IEEE, 2015.

[6] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-

direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.

[7] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2017.

[8] Paul Furgale, Pat Carle, John Enright, and Timothy D Barfoot. The devon island rover navigation dataset. *The International Journal of Robotics Research*, 31(6):707–713, 2012.

[9] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1280–1286. IEEE, 2013.

[10] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. Ieee, 2011.

[11] Chao X Guo and Stergios I Roumeliotis. Imu-rgbd camera extrinsic calibration: Observability analysis and consistency improvement. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Karlsruhe, Germany*, pages 2920–2927, 2013.

[12] Hailin Jin, Paolo Favaro, and Stefano Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 684–689. IEEE, 2001.

[13] Julian Jordan and Andreas Zell. Ground plane based visual odometry for rgbd-cameras using orthogonal projection. *IFAC-PapersOnLine*, 49(15): 108–113, 2016.

[14] 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.

[15] Christian Kerl. Odometry from rgb-d cameras for autonomous quadrocopters. *Master's Thesis, Technical University*, 2012.

[16] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3748–3754. IEEE, 2013.

[17] Pyojin Kim, Hyon Lim, and H Jin Kim. Robust visual odometry to irregular illumination changes with rgb-d camera. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3688–3694. IEEE, 2015.

[18] Pyojin Kim, Brian Coltin, Oleg Alexandrov, and H Jin Kim. Robust visual localization in changing lighting conditions. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5447–5452. IEEE, 2017.

[19] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 486–492. IEEE, 2010.

[20] Sebastian Klose, Philipp Heise, and Alois Knoll. Efficient compositional

approaches for real-time robust direct visual odometry from rgb-d data. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1100–1106. IEEE, 2013.

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

[22] Mingyang Li and Anastasios I Mourikis. Vision-aided inertial navigation for resource-constrained systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1057–1063. IEEE, 2012.

[23] 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.

[24] Mingyang Li and Anastasios I Mourikis. Optimization-based estimator design for vision-aided inertial navigation. In *Robotics: Science and Systems*, pages 241–248, 2013.

[25] Mingyang Li and Anastasios I Mourikis. Online temporal calibration for camera–imu systems: Theory and algorithms. *The International Journal of Robotics Research*, 33(7):947–964, 2014.

[26] Mingyang Li, Hongsheng Yu, Xing Zheng, and Anastasios I Mourikis. High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 409–416. IEEE, 2014.

[27] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[28] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Robotics and automation, 2007 IEEE international conference on*, pages 3565–3572. IEEE, 2007.

[29] Anastasios I Mourikis, Nikolas Trawny, Stergios I Roumeliotis, Andrew E Johnson, Adnan Ansar, and Larry Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, 2009.

[30] Vo Quang Nhat and Gueesang Lee. Illumination invariant object tracking with adaptive sparse representation. *International Journal of Control, Automation and Systems*, 12(1):195–201, 2014.

[31] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. Ieee, 2004.

[32] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[33] Vasko Sazdovski and Peter MG Silson. Inertial navigation aided by vision-based simultaneous localization and mapping. *IEEE Sensors Journal*, 11 (8):1646–1656, 2011.

[34] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011.

[35] Jianbo Shi and Carlo Tomasi. Good features to track. Technical report, Cornell University, 1993.

[36] Joan Sola. Consistency of the monocular ekf-slam algorithm for three different landmark parametrizations. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3513–3518. IEEE, 2010.

[37] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Real-time visual odometry from dense rgb-d images. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 719–722. IEEE, 2011.

[38] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.

[39] Ke Sun, Kartik Mohta, Bernd Pfrommer, Michael Watterson, Sikang Liu, Yash Mulgaonkar, Camillo J Taylor, and Vijay Kumar. Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, 3(2):965–972, 2018.

[40] Zhen Tian, Jian Li, Qing Li, and Nong Cheng. A visual-inertial navigation system based on multi-state constraint kalman filter. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2017 9th International Conference on*, volume 1, pages 199–202. IEEE, 2017.

[41] Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. Direct

visual-inertial odometry with stereo cameras. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1885–1892. IEEE, 2016.

[42] Kejian J Wu, Ahmed M Ahmed, Georgios A Georgiou, and Stergios I Roumeliotis. A square root inverse filter for efficient vision-aided inertial navigation on mobile devices. In *2015 Robotics: Science and Systems Conference, RSS 2015*. MIT Press Journals, 2015.

[43] Xing Zheng, Zack Moratto, Mingyang Li, and Anastasios I Mourikis. Photometric patch-based visual-inertial odometry. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3264–3271. IEEE, 2017.

# Appendix A

# Derivation of Photometric Error Jacobian

In this appendix, Jacobian matrices of the photometric error are derived. Define the following variables,

$$\mathbf{g} = \begin{bmatrix} X' & Y' & Z' \end{bmatrix}^{\mathrm{T}} \tag{A.1}$$

which is a warped 3D feature position.

$$\mathbf{\Pi}^{-1}(\mathbf{x}_i) = \begin{bmatrix} X & Y & Z \end{bmatrix}^{\mathrm{T}} \tag{A.2}$$

which is a 3D feature position before warping. These positions are related by the rigid body transformation matrix $\mathbf{T} \in \mathbb{SE}(3)$ such that,

$$\mathbf{g} = \mathbf{R}\mathbf{\Pi}^{-1}(\mathbf{x}_i) + \mathbf{t} \tag{A.3}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.4}$$

Also, define a vectorized transformation matrix as below,

$$\mathbf{T}^* \triangleq \begin{bmatrix} r_{11} & r_{21} & r_{31} & r_{12} & r_{22} & \cdots & t_1 & t_2 & t_3 \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{12} \tag{A.5}$$

The first derivative in Eq. (3.18) is an image gradient at the warped image point,

$$\left.\frac{\partial I_2}{\partial \mathbf{\Pi}}\right|_{\mathbf{\Pi}_k} = \nabla I_2(\mathbf{\Pi}_k) \tag{A.6}$$

The second derivative in Eq. (3.18) is derived as below,

$$\left.\frac{\partial \mathbf{\Pi}}{\partial \mathbf{g}}\right|_{\mathbf{g}_k} = \begin{bmatrix} \frac{f_u}{Z'} & 0 & -\frac{f_u X'}{Z'^2} \\ 0 & \frac{f_v}{Z'} & -\frac{f_v Y'}{Z'^2} \end{bmatrix} \tag{A.7}$$

where $f_u$ and $f_v$ are vertical and horizontal focal length, respectively. According to Eq. A.3, the warped feature position can be written as below,

$$\mathbf{g} = \begin{bmatrix} r_{11}X + r_{12}Y + r_{13}Z + t_1 \\ r_{21}X + r_{22}Y + r_{23}Z + t_2 \\ r_{31}X + r_{32}Y + r_{33}Z + t_3 \end{bmatrix} \tag{A.8}$$

The third derivative in Eq. (3.18) is,

$$\left.\frac{\partial \mathbf{g}}{\partial \mathbf{T}^*}\right|_{\mathbf{T}_k} = \begin{bmatrix} X\mathbf{I}_3 & Y\mathbf{I}_3 & Z\mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix} \tag{A.9}$$

Note that the denominator of Eq. (A.9) is not $\mathbf{T}$ but $\mathbf{T}^*$ to avoid tensor notations.

Lastly, the fourth derivative in Eq. (3.18) is,

$$\frac{\partial \mathbf{T}^*}{\partial \boldsymbol{\xi}} = \begin{bmatrix} 0 & r_{31} & -r_{21} & 0 & 0 & 0 \\ -r_{31} & 0 & r_{11} & 0 & 0 & 0 \\ r_{21} & -r_{11} & 0 & 0 & 0 & 0 \\ 0 & r_{32} & -r_{22} & 0 & 0 & 0 \\ -r_{32} & 0 & r_{12} & 0 & 0 & 0 \\ r_{22} & -r_{12} & 0 & 0 & 0 & 0 \\ 0 & r_{33} & -r_{23} & 0 & 0 & 0 \\ -r_{33} & 0 & r_{13} & 0 & 0 & 0 \\ 0 & t_1 & -t_2 & 1 & 0 & 0 \\ -t_1 & 0 & -t_3 & 0 & 1 & 0 \\ t_2 & t_3 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.10}$$

# 국문초록

본 논문에서는 로버 항법 시스템을 위해 관성측정장치와 스테레오 카메라를 사용하여 빛 변화에 강건한 직접 방식 영상 오도메트리와 자가 보정 영상관성 항법 알고리즘을 제안한다. 기존 대부분의 영상기반 항법 알고리즘들은 램버션 표면 가정을 위배하는 야외의 강한 햇빛 혹은 일정하지 않은 카메라의 노출 시간으로 인해 영상의 밝기 변화에 취약하였다. 한편, 영상 오도메트리의 오차 누적을 줄이기 위해 관성측정장치를 사용할 수 있지만, 영상관성 시스템에 대한 외부 교정 변수는 공간 및 시간적으로 영상 및 관성 좌표계를 연결하기 때문에 사전에 정확하게 계산되어야 한다. 본 논문은 로버 항법을 위해 지역 및 전역적인 빛 변화를 설명하는 직접 방식 영상 오도메트리의 버킷 밝기 모델을 제안한다. 또한, 본 연구에서는 스트레오 카메라에서 측정된 특징점을 이용하여 관성측정장치와 카메라간의 시간 오프셋과 상대 위치 및 자세를 추정하는 자가 보정 영상관성 항법 알고리즘을 제시한다. 특히, 제안하는 영상관성 알고리즘은 확장 칼만 필터에 기반하며 교정 파라미터를 필터의 상태변수에 확장하였다. 제안한 직접방식 영상 오도메트리는 달 유사환경에서 촬영된 오픈소스 데이터셋을 통해 그 성능을 검증하였다. 또한 상용 센서 및 로버 플랫폼을 이용하여 테스트 로버를 설계하였고, 이를 통해 영상관성 시스템을 자가 보정 할 경우 그렇지 않은 경우 보다 회기 위치 오차(return position error)가 76.4% 감소됨을 확인하였다.

**주요어**: 로버 항법, 직접 방식 영상 오도메트리, 영상관성 항법, 자가 보정
**학번**: 2017-25371