Universität des Saarlandes
Max-Planck-Institut für Informatik

Keyframe-based Visual-Inertial Odometry for Small Workspace

Masterarbeit im Fach Informatik
Master's Thesis in Computer Science
von / by
Xi Li

angefertigt unter der Leitung von / supervised by
Dr. Roland Angst

betreut von / advised by
Dr. Roland Angst

begutachtet von / reviewers
Dr. Roland Angst
Prof. Dr. Antonio Krüger

Saarbrücken, June 2016

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement in Lieu of an Oath

I hereby confirm under oath that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass die vorliegende Arbeit mit der elektronischen Version übereinstimmt.

Statement in Lieu of an Oath

I hereby confirm the congruence of the contents of the printed data and the electronic version of the thesis.

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

(Datum / Date) (Unterschrift / Signature)

Acknowledgments

Keyframe-based Visual-Inertial Odometry for Small Workspace

by

Xi Li

Submitted to the Department of Computer Science
on June 1, 2016, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science

Abstract

Thesis Supervisor: Roland Angst
Title: Dr.

Contents

Chapter 1

Introduction

In past few years, the development of *Robotics* has surpassed people's expectation. The word *Robotics* has been first appeared in science fiction "Liar!" by Issac Asimov [30], it referred to science and technology of robots. By definition, *Robotics* is a research branch that related to design, control and application of robots, as well as processing feedback from robots.

Modern robots have been classified into several categories(e.g., Mobile robot, industrial robot, service robot, education robot etc.) with their usages. Among those categories, full-autonomous or semi-autonomous mobile robots attracts more and more researches. Such robots have abilities to move around in their moving space, with or without humans' control. The aim of research in mobile robots is to help us accomplish various hard tasks, whether domestically, commercially or militarily. These tasks, such as assisting disabled people, defusing bombs, or repair equipment in dangerous place is either risky or high expense for human beings.

For mobile robot, finding physical location of itself in unknown environment is normally crucial, such an ability(e.g., *Robot Navigation*) allows mobile robots avoid risky obstacles and finally arrive the goal position. Roughly speaking, *Robot Navigation* is a computing system which processes the information from external sources(e.g., sensors) and apply an algorithm to navigate robot, and sometimes build a map of environment. In *Robot Navigation*, robots sense environmental information by their sensors. These sensors, either locally (e.g., camera, inertial measurement unit (IMU)), or globally (e.g., Global Positioning System) detect events or changes in environment, and transfer their data to robots. Generally, sensors equipped in mobile robot (Local Sensor) are designed light, small, and inexpensive considering convenient movement and low expense. Camera and IMU sensor are considered most common local sensor in small mobile robot system.

A camera is a optical instrument for capturing images. Modern camera has several advantages for *Robot Navigation*. First, the core of camera chip set is cheap and easily installed in any mobile robot system; Second, camera often brings very rich information as it simulates the functioning of human eye. By recognizing key-points [24, 18, 23] in certain images, system observes the *landmarks* in environment, and those *landmarks* will localize robots by *Triangulation* [4, 14, 6].

A IMU sensor (Figure 1-1) often combines *gyroscope* and *accelerometer*, sometimes

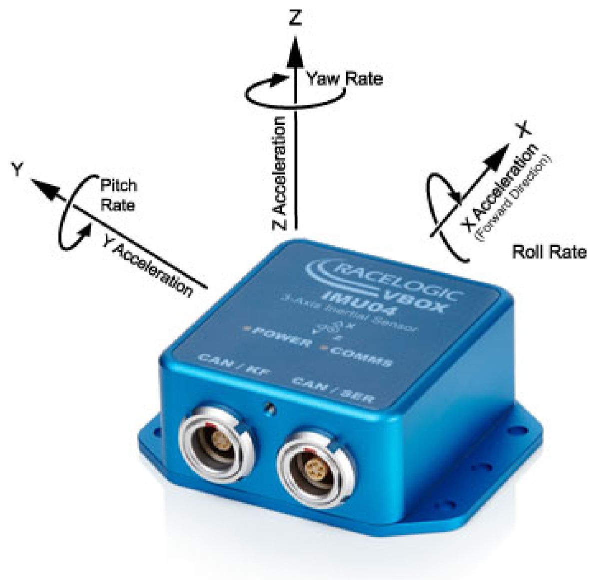


Figure 1-1: IMU sensor with gyroscope and accelerometer measures rotational rate and acceleration of X, Y, Z axis regarding its local frame. Note that IMU sensor normally has bias and irreducible noises, it is necessary to apply a calibration like cameras. The frequency of IMU output is normally larger than 100 Hz. Source: [1]

magnetometer, to measure specific force, angular rate and magnetic field regarding to its local frame. IMU is one of main component in *Inertial Navigation System*, which firstly used in air plane, spacecraft, guided missiles, and now also in mobile robot [2, 15, 20, 8]. IMU sensor utilize *Dead Reckoning* to track device's position, such technology tries to integrate IMU data over time by assuming the movement model of devices fixed(i.e., acceleration and rotation rate is constant over small period of time).

1.1 Motivation and Contribution

The main motivation of this thesis is to improve the navigation accuracy by fusing camera data and IMU sensor data.

Single sensor-based navigation system may not satisfy the requirements of high-quality localization by mobile robot. GPS-based navigation system has been used for outdoor devices for long time. However, such a system suffered from localizing in in-door environment, and also the accuracy of general GPS is not high, i.e., 3 to 5 meters error [29]. For mobile robot, which may move from in-door to out-door, and requires high-accuracy navigation, GPS is mostly not used, or as baseline of navigation [12].

Vision-based navigation system [4, 14], or simultaneous localization and mapping (SLAM) [5, 7, 21] gives an acceptable navigation result. [4] recognizes corner feature by single camera, and it uses a extended kalman filter framework to track the uncertainty and propagates the system state, however it can not handle large-scale scene. [14] utilizes key-frame based bundle adjustment to update the map, improves both accuracy and efficiency, it still met some problem in large-scale scene, becasue vision-based method often is a trade-off between computational complexity and localization accuracy due to the rich information and low output frequency by camera.

Single *Inertial Navigation System* recognize its pose by *Dead Reckoning* [19, 16]. However, the problem of *Dead Reckoning* error accumulation; Only few directions are observable [12] by IMU during whole navigation process. When object corrupts movement assumption, a correction step by external data will be needed.

Fusing camera and IMU data has many advantages. On the one hand, it can decrease computational time by making good use of high-frequency IMU data; On the other hand, it can reduce the error accumulation of IMU integration by the correction of vision-based navigation result within several turn. Fuse the camera and IMU data for robot navigation is not novel. [20] applies multi-state kalman filter (MSCKF) on state update, decrease the computational complexity by only keeping few last information of keyframe. [12] analysis the consistency of MSCKF, correct the ways of IMU integration to obtain consistency of sytem, therefore increase the accuracy. [8] exploits a way to optimize manifold information, therefore solving data fusing problem in a non-linear optimization scheme. Unfortunately, codes and data of above methods are not accessible, that motivates to explore possible methods to increase accuracy of *vision-inertial navigation system* both theoretically and experimentally.

The contributions of this thesis are as follows,

- We exploit a highly flexible, realistic software to generate synthetic IMU sensor data and corresponding vision data, that is the main source to provide experimental data in this thesis.
- We present error-state kalman filter system kinematic based on quaternion, explore the multiple ways to integrate IMU data due to different movement models.
- We propose a novel method to update fusing result based on key-framed bundle adjustment.
- We propose a real-time visual-inertial framework implemented in C++, which is stable, scalable, high-accuracy and low-latency.

1.2 Outline of the Thesis

In Chapter 2 we first overview our visual-inertial odometry, including world representation, important notations, we also compare filter method and key-frame based method in SLAM problem, and in the end we explain our choice in this thesis.

Then we enter the Chapter 3, which introduces *Quaternion Algebra*. In this chapter, we introduces the basic operations on quaternion, the relationship among quaternion, rotation matrix and rotation vector. In this chapter, we also explain how to integrate or derivative quaternion over time.

Chapter 4 is main part of this thesis. In Section 4.1, we study the Error-State Kalman Filter (ESKF), and apply it to IMU integration; Section 4.2 we summarize how to fuse camera into ESKF, and how to optimize it by key-frame based bundle adjustment. In Section 4.3, we overview the pipeline of our visual-inertial odometry system.

We show our experiment results in Chapter 5. First, the process of generating synthetic IMU and camera data is presented. Then we run several experiments to show our proposed visual-inertial odometry has higher accuracy than single IMU integration, visual SLAM, as our system is still running in real-time.

In the end, we summarize and discuss our work and analysis the potential future work in Chapter 6.

Chapter 2

Overview of Visual-inertial Odometry

In this chapter, we will overview visual-inertial odometry system. In section 2.1, we will introduce world representation(e.g., world frame, camera frame and IMU frame) together with basic notations in our odometry system. Then in section 2.2, we will discuss two important scheme, filter method and keyframe Bundle Adjustment(keyframe BA) in SLAM algorithm, and explain why we finally choose keyframe-based method.

2.1 World Representations and Notations

Visual-inertial odometry [17], literally, is an odometry system, which received environment information by visual (camera) and inertial (IMU) sensor. VIO is similar with well-known visual odometry (VO) problem [22], with an additional IMU sensor, it tries to estimate agent's pose as agent keep moving in the environment. One big difference between VIO and SLAM algorithm is that VIO does not or only build a simple map, whereas SLAM normally maintains and continuously updates a map.

To setup a VIO system, we need to first define the ways to represent the world. Globally, we have a world frame $\mathcal{W}$; World frame $\mathcal{W}$ is set to a right-handed Cartesian coordinate system that every objects has an absolute pose (translation and rotation) in it. Then we have local frame for each sensor, i.e., IMU frame $\mathcal{I}$ and camera frame $\mathcal{C}$; Every time camera and IMU sensor obtain observations within their own local frame, we need to integrate those data and estimate the pose of those sensors in world frame $\mathcal{W}$. Figure 2-1 shows the overall world representations. Both IMU frame and Camera frame are right-handed Cartesian coordinate system.

In this master thesis, we use following notations,

- We denote scalars as a, b, c, vectors as $\mathbf{a}, \mathbf{b}, \mathbf{c}$, matrices as $\mathbf{A}, \mathbf{B}, \mathbf{C}$, frames as $\mathcal{A}, \mathcal{B}, \mathcal{C}$.

- We denote measurement $\mathbf{m}$ in particular frame $\mathcal{F}$ as $\mathbf{m}_{\mathcal{F}}$. To further simplify, any parameter that is **not** in world frame shall be denoted particularly. For example, the translation $\mathbf{p}$ in camera frame will be denoted as $\mathbf{p}_{\mathcal{C}}$, and the translation $\mathbf{p}$ in world frame will be denoted as $\mathbf{p}$.

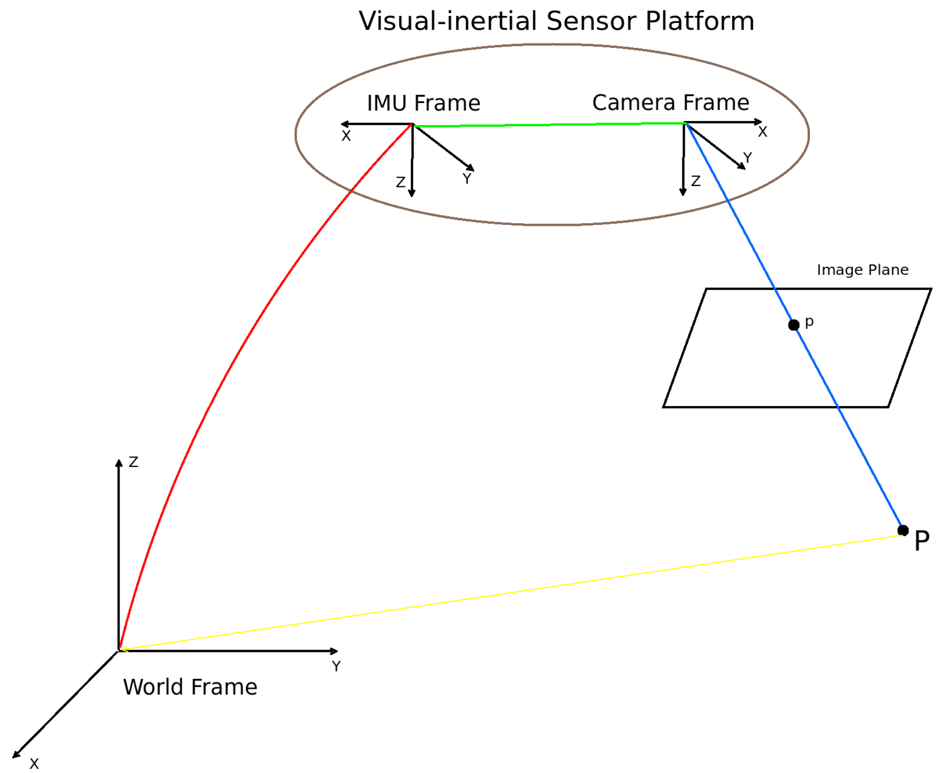


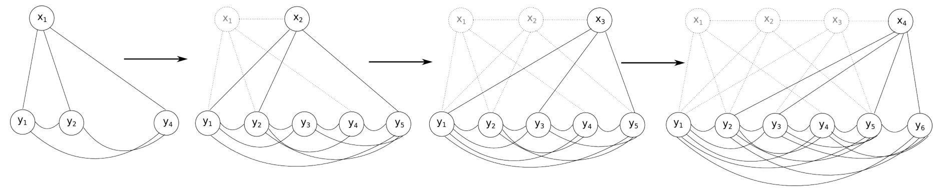
Figure 2-1: This figure shows the connection among world frame $\mathcal{W}$, IMU frame $\mathcal{I}$ and camera frame $\mathcal{C}$. Green line shows the transformation between camera and IMU, which can be pre-calibrated. Red line is the pose of IMU in world frame. Camera frame observes point $\mathbf{p}$ of object P in image plane, and connects a object P by blue line, and the coordinate of object P in world frame is presented as yellow line.

- A general translation $\mathbf{t}$ should express a translation from point A to point B in frame $\mathcal{C}$, which is denoted as $\mathbf{t}_{\mathcal{C}}^{AB}$. We simplify a point $\mathbf{p}$ in frame $\mathcal{A}$ as $\mathbf{p}_{\mathcal{A}}$, when this point is the translation $\mathbf{t}_{\mathcal{A}}^{OP}$, O is origin of frame $\mathcal{A}$, and $\mathbf{p} = P$, this holds same for vector.

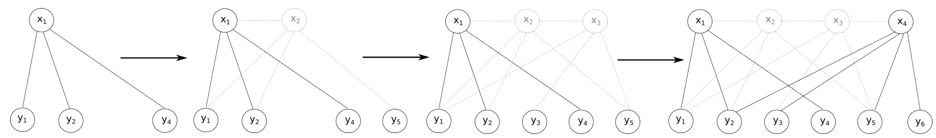
- A general rotation is either expressed in quaternion $\mathbf{q}$ or rotation matrix $\mathbf{R}$. We use quaternion $\mathbf{q}$ as example. A quaternion is a orientation operation from frame $\mathcal{B}$ to frame $\mathcal{A}$, and it is denoted as $\mathbf{q}_{\mathcal{AB}}$ in this thesis. Noted that if such a operation is from world frame $\mathcal{W}$ to some frame $\mathcal{B}$, we omit both frame for simplification, i.e., $\mathbf{q}_{\mathcal{WB}} \triangleq \mathbf{q}$.

2.2 Filter Versus Keyframe

It is important to note that though this thesis focus on visual-inertial odometry for small workspace, we still tend to keep the possibility to extend our system to a general, scalable and efficient SLAM system. SLAM system usually have two parallel process, one is for localization and the other is for mapping, the crucial point of building such



(a) Filter



(b) Keyframe-based BA

Figure 2-2: (a) Filter method for SLAM. (b) Keyframe-based Bundle Adjustment (BA) for SLAM. We denote the i^{th} camera position as $\mathbf{x}_i$, i^{th} image feature as $\mathbf{y}_i$. We connect the line between camera and image feature if this feature is observed by this camera, the vanished observations is presented as dotted line, and the vanished camera is expressed as grey font. Both graph changes as time goes on from left to right. One can see from (a) that though only the latest camera pose is reserved, the edges between features are increasing exponentially. (b) stores some of former camera poses (keyframe) (i.e., $\mathbf{x}_1$ and $\mathbf{x}_4$) by keeping graph stay sparsity.

a system is to keep both processes efficient. There are two general framework (e.g., filter-based method and keyframe-based method) in SLAM. In this section, we want to discuss whether filter-based method or keyframe-based method are more suitable for our case.

Filter-based SLAM [5, 6, 4] uses *Extended Kalman Filter* (EKF) to propagate state and update the covariance of the state. In each step, system obtain the current pose estimation and map update by marginalising all former information. This marginalising step usually eliminates the former pose and adds connections to image features. As showed in Figure 2-2a, the graph will not grow fast with time since the former pose has been eliminated and features in environment is limited. However, once the system moves to large scale scene, the problem of limiting the number of features become severe as the graph tends to be fully-connected.

Keyframe-based SLAM [14, 20, 9, 7, 21] applies *bundle adjustment* (BA) for keyframes to update the map in each step. In keyframe-based SLAM, it stores some historical poses (keyframes), and combines with image feature points to do a BA step. The chosen of keyframes varies from implementations, the idea is to pick up the poses that is not very close to last keyframe, otherwise the information might be redundant, which might lead to increase the computational cost. In Figure 2-2b, the graph still stays sparsity as the number of poses and features increases. The drawback might be the behaviour of pose estimation is inadequate as it ignores some of former information.

In [25], they conclude that keyframe-based SLAM is slightly better than filter-based SLAM in their experiment settings, especially when scale of scene becomes larger. In this master thesis, we choose keyframe-based method for visual part and filter method for IMU integration part. We choose filter method for IMU integration part is that we do not keep former information (e.g., image features or landmarks) in integration step, hence each filter step can be regarded as an optimization step. The reason why we use keyframe-based method for visual part is that we want keep the scalability of our system, besides the results from IMU integration can be a good compensation in case of the lack of pose estimation in keyframe-based BA.

Chapter 3

Background on Quaternion Algebra

One important task for this master thesis is to integrate IMU data over time to estimate camera pose (e.g., position and orientation). By assuming movement model, it is straightforward to integrate position in Cartesian space, however integrating orientation in manifold space is often not a easy task. In this chapter, we introduce the quaternion algebra, and explore the way to operate quaternion over time.

3.1 Definition of Quaternion

A quaternion Q is defined as,

$$Q = q_w + q_x i + q_y j + q_z k \tag{3.1}$$

where $\{q_w, q_x, q_y, q_z\} \in \mathbb{R}$, and $\{i, j, k\}$ are three imaginary unit length, e.g., $i^2 = j^2 = k^2 = ijk = -1$.

In most cases, we represent quaternion Q as a four-element vector $\mathbf{q}$ composed by above four real number $\{q_w, q_x, q_y, q_z\}$, i.e.,

$$\mathbf{q} \triangleq [q_w \ \mathbf{q}_v]^T = [q_w \ q_x \ q_y \ q_z]^T \tag{3.2}$$

where q_w is the real part of $\mathbf{q}$, and q_v is a 3-vector to represent imaginary part of $\mathbf{q}$.

It is worth to be noted that there are two different conventions of quaternion $\mathbf{q}$, *Hamilton way* [11] and *JPL way* [3]. In Hamilton convention, the real part q_w is the first component of $\mathbf{q}$, i.e., $[q_w \ \mathbf{q_v}]$, whereas in JPL way, the real part is the fourth component, i.e., $[\mathbf{q_v} \ q_w]$. To avoid confusions, and considering Hamilton way is more common to use, especially for implementation [10, 13], we hereby claim that we use ***Hamilton way*** to represent quaternion $\mathbf{q}$ throughout this master thesis.

3.2 Properties of Quaternion

In this section, we will introduce properties of quaternion.

Summation We start with summation of two quaternions $\mathbf{q}$ and $\mathbf{p}$,

$$\mathbf{q}+\mathbf{p}=[q_w+p_w\ \mathbf{q}_v+\mathbf{p}_v]^T=[q_w+p_w\ q_x+p_x\ q_y+p_y\ q_z+p_z\ ]^T \tag{3.3}$$

Product We use $\otimes$ to denote the product operation on quaternions, which gives,

$$\mathbf{q}\otimes\mathbf{p}=\begin{bmatrix}p_wq_w-p_xq_x-p_yq_y-p_zq_z\\ p_wq_x+p_xq_w+p_yq_z-p_zq_y\\ p_wq_y-p_xq_q+p_yq_w+p_zq_x\\ p_wq_z-p_xq_y-p_yq_z+p_zq_w\end{bmatrix} \tag{3.4}$$

The product of two quaternions can be expressed as two equivalent matrix products,

$$\mathbf{q_1}\otimes\mathbf{q_2}=Q_1^+\mathbf{q_2} \tag{3.5}$$

with

$$Q_1^+=q_w\mathbb{I}+\begin{bmatrix}0 & -\mathbf{q_v}^T\\ \mathbf{q_v} & [\mathbf{q_v}]_\times\end{bmatrix} \tag{3.6}$$

where $[\mathbf{q}_v]_\times$ represents the cross-product matrix of $\mathbf{q}_v$, which the cross-product matrix of a vector $\mathbf{v}$ is defined by

$$[\mathbf{v}]_\times=\begin{bmatrix}0 & -v_z & v_y\\ v_z & 0 & -v_x\\ -v_y & v_x & 0\end{bmatrix} \tag{3.7}$$

note that the quaternion product is not commutative, i.e.,

$$\mathbf{p}\otimes\mathbf{q}\neq\mathbf{q}\otimes\mathbf{p} \tag{3.8}$$

however it is associative, and distributive over sum, i.e.,

$$\mathbf{p}\otimes(\mathbf{q}\otimes\mathbf{k})=(\mathbf{p}\otimes\mathbf{q})\otimes\mathbf{k} \tag{3.9}$$

$$\mathbf{p}\otimes(\mathbf{q}+\mathbf{k})=\mathbf{p}\otimes\mathbf{q}+\mathbf{p}\otimes\mathbf{k} \tag{3.10}$$

$$(\mathbf{q}+\mathbf{k})\otimes\mathbf{p}=\mathbf{q}\otimes\mathbf{p}+\mathbf{k}\otimes\mathbf{p} \tag{3.11}$$

Conjugate The conjugate $\mathbf{q}^*$of a quaternion is defined by

$$\mathbf{q}^*\triangleq q_w-\mathbf{q_v}=[q_w\ \ -\mathbf{q_v}]^T \tag{3.12}$$

Identity We call a quaternion $\mathbf{q}$ identical if, for any given quaternion $\mathbf{p}$, $\mathbf{q}\otimes\mathbf{p}=\mathbf{p}\otimes\mathbf{q}=\mathbf{p}$. An identical quaternion $\mathbf{q}$ satisfies that,

$$\mathbf{q}=[1\ 0\ 0\ 0]^T \tag{3.13}$$

In this master thesis, we denote identical quaternion as $\mathbf{q}_\mathbb{I}$.

Norm The norm of a quaternion $\|\mathbf{q}\|$ is defined similar to the norm of a vector, which is,

$$\|\mathbf{q}\| = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \tag{3.14}$$

Inverse The inverse of a quaternion $\mathbf{q}^{-1}$ is defined as,

$$\mathbf{q}^{-1} = \mathbf{q}^* / \|\mathbf{q}\| \tag{3.15}$$

which leads to,

$$\mathbf{q}^* \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q}^* = \mathbf{q}_{\mathbb{I}} \tag{3.16}$$

Unit quaternion The norm of a unit quaternion $\|\mathbf{q}\|$ is 1, and the inverse of such a unit quaternion is equal to the conjugate of this quaternion,

$$\mathbf{q}^{-1} = \mathbf{q}^* \tag{3.17}$$

Pure quaternion A pure quaternion $\mathbf{q}$ is defined as,

$$\mathbf{q} = [0\ \mathbf{q}_v]^T = [0\ q_x\ q_y\ q_z]^T \tag{3.18}$$

Let pure quaternion $\mathbf{q} = \theta\mathbf{u}$, where $\theta = \|\mathbf{q}\|$, we can compute the exponential of $\mathbf{q}$ with the help of Euler formula,

$$\mathrm{e}^{\mathbf{q}} = \mathrm{e}^{\theta\mathbf{u}} = \cos\theta + \mathbf{u}\sin\theta = [\cos\theta\ \mathbf{u}\sin\theta]^T \tag{3.19}$$

which is still a quaternion, and moreover $\mathrm{e}^{\mathbf{q}}$ is a unit quaternion because its norm $\|\mathrm{e}^{\mathbf{q}}\|^2 = \cos\theta^2 + \sin\theta^2 = 1$.

3.3 Quaternions and Rotation operations

We discuss the relationship between quaternions and rotation operations by first introducing rotation vector $\mathbf{v}$.

Given a rotation vector $\mathbf{v} = \phi\mathbf{u}$, where ϕ is the norm of $\mathbf{v}$ and $\mathbf{u}$ is a unit vector, we can rotate a vector $\mathbf{x}$ by an angle ϕ around the axis $\mathbf{u}$ following right-handed rule, and obtain a new vector $\mathbf{x}'$,

$$\mathbf{x}' = \mathbf{x}_{||} + \mathbf{x}_{\perp}\cos\phi + (\mathbf{u} \times \mathbf{x})\sin\phi \tag{3.20}$$

where $\mathbf{x}_{||} = (\mathbf{x} \cdot \mathbf{u})\mathbf{u}$ is the component parallel to $\mathbf{x}$, and $\mathbf{x}_{\perp} = -\mathbf{u} \times (\mathbf{u} \times \mathbf{x})$ is the component perpendicilar to $\mathbf{x}$, therefore $\mathbf{x} = \mathbf{x}_{||} + \mathbf{x}_{\perp}$. This formula is known as *vector rotation formular* or *Rodrigues formula.*

We then can define a rotation matrix $\mathbf{R}$ by rotation vector $\mathbf{v} = \phi\mathbf{u}$ by the help of Equation (3.7) as,

$$\mathbf{R} = \mathrm{e}^{[\mathbf{v}]_\times} \tag{3.21}$$

we can rotate a vector $\mathbf{x}$ by an angle ϕ around the axis $\mathbf{u}$ using $\mathbf{R}$ in a clean way,

$$\mathbf{x}' = \mathbf{R}\mathbf{x} \tag{3.22}$$

one can show that result in Equation (3.22) is equivalent with the result in Equation (3.20) [28].

Constructing a unit quaternion $\mathbf{q}$ by Equation (3.19) with a rotation vector $\mathbf{v} = \phi\mathbf{u}$,

$$\mathbf{q} = \mathrm{e}^{\mathbf{v}/2} = \begin{bmatrix} \cos\phi/2 \\ \mathbf{v}\sin\phi/2 \end{bmatrix} \tag{3.23}$$

we can rotate a vector $\mathbf{x}$ by an angle ϕ around the axis $\mathbf{u}$ by,

$$\mathbf{x}' = \mathbf{q} \otimes \mathbf{x} \otimes \mathbf{q}^* \tag{3.24}$$

we then show $\mathbf{x}'$ in Equation (3.24) is equivalent with $\mathbf{x}'$ in Equation (3.20).

We transferred the vector $\mathbf{x}$ into pure quaternion form as,

$$\mathbf{x}_q = [0\ \mathbf{x}]^T \tag{3.25}$$

then we rewrite formula (3.24) as,

$$\begin{bmatrix} 0 \\ \mathbf{x}' \end{bmatrix} = \begin{bmatrix} \cos\phi/2 \\ \mathbf{v}\sin\phi/2 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} \otimes \begin{bmatrix} \cos\phi/2 \\ -\mathbf{v}\sin\phi/2 \end{bmatrix} \tag{3.26}$$

expanding it by Equation (3.4), it is easily to show that,

$$\mathbf{x}' = \mathbf{x}_{||} + \mathbf{x}_{\perp}\cos\phi + (\mathbf{u} \times \mathbf{x})\sin\phi \tag{3.27}$$

which is exactly Equation (3.20).

To summarize here, we can construct a quaternion $\mathbf{q}$ or a rotation matrix $\mathbf{R}$ by any rotation vector $\mathbf{v} = \phi\mathbf{u}$, we denote such a quaternion as $\mathbf{q}\{\mathbf{v}\}$ and rotation matrix as $\mathbf{R}\{\mathbf{v}\}$ respectively. And a rotation operation of a vector $\mathbf{x}$ related to $\mathbf{v}$ can either be expressed as quaternion $\mathbf{q}\{\mathbf{v}\} \otimes \mathbf{x} \otimes \mathbf{q}\{\mathbf{v}\}^*$, or a rotation matrix $\mathbf{R}\{\mathbf{v}\}\mathbf{x}$. Note that we sometimes simplify $\mathbf{R}\{\mathbf{v}\}$ to $\mathbf{R}$, and/or $\mathbf{q}\{\mathbf{v}\}$ to $\mathbf{q}$ in this master thesis.

We also give conversion from rotation matrix $\mathbf{R}$ to quaternion $\mathbf{q}$. Knowing that,

$$\mathbf{q} \otimes \mathbf{x} \otimes \mathbf{q}^* = \mathbf{R}\mathbf{x} \tag{3.28}$$

we can construct $\mathbf{R} = \mathbf{R}\{\mathbf{q}\}$ by,

$$\mathbf{R} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{3.29}$$

and a conversion from quaternion to rotation matrix can be found in [27].

3.4 Time-derivatives on Quaternion

We introduce the time-derivative on quaternion by first define,

$$\mathbf{q}(t+\Delta t) \triangleq \mathbf{q}(t) \otimes \Delta \mathbf{q} \tag{3.30}$$

where $\mathbf{q}(t)$ is the quaternion at time t and $\Delta \mathbf{q}$ is quaternion transformation within a small period time Δt.

One can expand $\Delta \mathbf{q}$ by Taylor expansions with Equation (3.23) to,

$$\Delta \mathbf{q} = \begin{bmatrix} 1 \\ \frac{1}{2}\Delta\theta \end{bmatrix} + O(\|\Delta\theta\|^2) \tag{3.31}$$

where $\Delta\theta$ is a angular vector corresponding to $\Delta \mathbf{q}$. In fact, the angular rate ω at time t is defined as,

$$\omega(t) \triangleq \lim_{\Delta t \to 0} \frac{\Delta\theta}{\Delta t} \tag{3.32}$$

which is one of measurements we can obtain from IMU sensor.

By definition of the derivative, we can obtain the time-derivative $\dot{\mathbf{q}}$ of quaternion $\mathbf{q}$ as,

$$\dot{\mathbf{q}} = \frac{d\mathbf{q}(t)}{dt} \triangleq \lim_{\Delta t \to 0} \frac{\mathbf{q}(t+\Delta t) - \mathbf{q}(t)}{\Delta t} \tag{3.33}$$

which follows,

$$\begin{aligned} \dot{\mathbf{q}} &\triangleq \lim_{\Delta t \to 0} \frac{\mathbf{q}(t+\Delta t) - \mathbf{q}(t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{\mathbf{q} \otimes \Delta\mathbf{q} - \mathbf{q}}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{\mathbf{q} \otimes (\begin{bmatrix} 1 \\ \frac{1}{2}\Delta\theta \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix})}{\Delta t} \\ &= \frac{1}{2}\mathbf{q} \otimes \begin{bmatrix} 0 \\ \omega \end{bmatrix} \end{aligned} \tag{3.34}$$

here we simplify $\mathbf{q}(t)$ to $\mathbf{q}$. Then we can obtain the time-derivative on quaternion by writing angular rate into pure quaternion form (3.18), which is,

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \omega \tag{3.35}$$

3.5 Time-integration on Quaternion

To integrate quaternion over time, we explore the relationship between $\mathbf{q}(t_n)$ and $\mathbf{q}(t_{n+1})$ where $t_n = n\Delta t$. Expanding $\mathbf{q}(t_{n+1})$ using Taylor series, we have

$$\mathbf{q}(t_{n+1}) = \mathbf{q}(t_n) + \dot{\mathbf{q}}(t_n)\Delta t + \frac{1}{2!}\ddot{\mathbf{q}}(t_n)\Delta t^2 + \frac{1}{3!}\dddot{\mathbf{q}}(t_n)\Delta t^3 + \dots \tag{3.36}$$

Assume that the second order derivative of rotational rate is zero, which is $\ddot{\omega} = 0$, we have

$$\dot{\mathbf{q}}(t_{n+1}) = \frac{1}{2}\mathbf{q}(t_n) \otimes \omega()(t_n) \tag{3.37}$$

$$\ddot{\mathbf{q}}(t_{n+1}) = \frac{1}{2^2}\mathbf{q}(t_n) \otimes \omega^2(t_n) + \frac{1}{2}\mathbf{q}(t_n) \otimes \dot{\omega} \tag{3.38}$$

$$\dddot{\mathbf{q}}(t_{n+1}) = \frac{1}{2^3}\mathbf{q}(t_n) \otimes \omega^3(t_n) + \frac{1}{4}\mathbf{q}(t_n) \otimes \dot{\omega}\omega(t_n) + \frac{1}{2}\mathbf{q}(t_n) \otimes \omega(t_n)\dot{\omega} \tag{3.39}$$

and so forth and so on. We then get the result of time integration by taking Equation (3.37, 3.38, 3.39) back into Equation (3.36).

We hereby gives a stronger assumption that angular rate $\omega(t_n)$ remains constant during a small time period Δt, which is $\dot{\omega} = 0$. However, considering the sampling rate of IMU sensor is usually high ($>$ 100 Hz), this assumption actually is general and also gives us a cleaner expression of time integration.

Given $\dot{\omega} = 0$, we can get

$$\mathbf{q}_{n+1} = \mathbf{q}_n \otimes (1 + \frac{1}{2}\omega_n\Delta t + \frac{1}{2!}(\frac{1}{2}\omega_n\Delta t)^2 + \frac{1}{3!}(\frac{1}{2}\omega_n\Delta t)^3 + \frac{1}{4!}(\frac{1}{2}\omega_n\Delta t)^4 + \dots) \tag{3.40}$$

here we regard the $\mathbf{q}$ and ω as series, which is exactly

$$\mathbf{q}_{n+1} = \mathbf{q}_n \otimes \mathrm{e}^{\omega\Delta t/2} \tag{3.41}$$

we can rewrite it by Equation (3.23) as,

$$\mathbf{q}_{n+1} = \mathbf{q}_n \otimes \mathbf{q}\{\omega_n\Delta t\} \tag{3.42}$$

which is called ***Zeroth order forward integration*** of quaternion over time.

We can obtain ***Zeroth order backward integration*** by assuming the angular rate remains ω_{n+1} within Δt, then we have

$$\mathbf{q}_{n+1} = \mathbf{q}_n \otimes \mathbf{q}\{\omega_{n+1}\Delta t\} \tag{3.43}$$

and ***Zeroth order midward integration*** by assuming the angular rate holds $\bar{\omega}_{n+1} = (\omega_n + \omega_{n+1})/2$ within Δt,

$$\mathbf{q}_{n+1} = \mathbf{q}_n \otimes \mathbf{q}\{\bar{\omega}_n\Delta t\} \tag{3.44}$$

Though not used in this master thesis, we notice that [26] gives ***First order integration*** by assuming angular rate is linear with time, i.e., $\dot{\omega} = \frac{\omega_{n+1} - \omega_n}{\Delta t}$, which is

$$\mathbf{q}_{n+1} = \mathbf{q}_n \otimes \mathbf{q}\{\bar{\omega}_n \Delta t\} + \frac{\Delta t^2}{48} \mathbf{q}_n \otimes (\omega_n) \otimes \omega_{n+1} - \omega_{n+1}) \otimes \omega_n) + \ldots \tag{3.45}$$

in our quaternion convention.

Chapter 4

Modular Sensor Fusing

4.1 Error-state Kalman Filter for IMU Integration

4.1.1 Motivation

4.1.2 System Kinematics

4.1.3 State Propagations

4.1.4 State Reset

4.2 Camera as Complementary Sensory Data

4.2.1 Motivation

4.2.2 Self-adapt Map Scale

4.2.3 Keyframe-based Bundle Adjustment

4.3 Visual-inertial Odometry Pipeline Overview

Chapter 5

Experiments

5.1 Synthetic Dataset

5.2 Some other experiments

Chapter 6

Summary, Discussion and Future Works

Appendix A

Integration Methods

A.1 Runge-Kutta Numerical Integration Methods

A.2 Closed-form Integration Methods

Appendix B

Approximation Methods

[20]

Bibliography

[1] Imu sensor https://www.vboxautomotive.co.uk/index.php/en/products/modules/inertial-measurement-unit, 2016. [Online; accessed 22-March-2016].

[2] Maxim A Batalin, Gaurav S Sukhatme, and Myron Hattig. Mobile robot navigation using a sensor network. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 636–641. IEEE, 2004.

[3] WG Breckenridge. Quaternions proposed standard conventions. *Jet Propulsion Laboratory, Pasadena, CA, Interoffice Memorandum IOM*, pages 343–79, 1999.

[4] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410. IEEE, 2003.

[5] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, 2007.

[6] Ethan Eade and Tom Drummond. Monocular slam as a graph of coalesced observations. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[7] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.

[8] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Robotics: Science and Systems (RSS)*, 2015.

[9] 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.

[10] G Guennebaud and B Jacob. The eigen c++ template library for linear algebra. *http:/eigen. tuxfamily. org*, 2010.

[11] William Rowan Hamilton. Ii. on quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(163):10–13, 1844.

[12] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *Robotics, IEEE Transactions on*, 30(1):158–176, 2014.

[13] Roland Hess. *The essential Blender: guide to 3D creation with the open source suite Blender*. No Starch Press, 2007.

[14] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[15] Taehee Lee, Joongyou Shin, and Dongil Cho. Position estimation for mobile robot using in-plane 3-axis imu and active beacon. In *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on*, pages 1956–1961. IEEE, 2009.

[16] Robert W Levi and Thomas Judd. Dead reckoning navigational system using accelerometer to measure foot impacts, December 10 1996. US Patent 5,583,776.

[17] Mingyang Li and Anastasios I Mourikis. Consistency of ekf-based visual-inertial odometry. *University of California Riverside, Tech. Rep*, 2011.

[18] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[19] BL McNaughton, Lijiang Chen, and EJ Markus. âĂIJdead reckoning,âĂİ landmark learning, and the sense of direction: a neurophysiological and computational hypothesis. *Cognitive Neuroscience, Journal of*, 3(2):190–202, 1991.

[20] Anastasios Mourikis, Stergios Roumeliotis, et al. 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.

[21] Raul Mur-Artal, JMM Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *Robotics, IEEE Transactions on*, 31(5):1147–1163, 2015.

[22] 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–652. IEEE, 2004.

[23] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):105–119, 2010.

[24] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[25] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Real-time monocular slam: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657–2664. IEEE, 2010.

[26] Nikolas Trawny and Stergios I Roumeliotis. Indirect kalman filter for 3d attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2:2005, 2005.

[27] JMP Van Waveren. From quaternion to matrix and back. *Id Software Inc*, 2005.

[28] Wikipedia. Rodrigues' rotation formula — wikipedia, the free encyclopedia, 2015. [Online; accessed 6-April-2016].

[29] Wikipedia. Global positioning system — wikipedia, the free encyclopedia, 2016. [Online; accessed 24-March-2016].

[30] Wikipedia. Robotics — wikipedia, the free encyclopedia, 2016. [Online; accessed 22-March-2016].