공학석사 학위논문

Pseudo-Outlier Method in Vision-Aided Navigation System Using Prior Information

영상보조 항법시스템에서의 사전 정보를 이용한 수도-아웃라이어 제거 기법

2020 년 2 월

서울대학교 대학원

기계항공공학부

정 재 영

Pseudo-Outlier Method in Vision-Aided Navigation System Using Prior Information

영상보조 항법시스템에서의 사전 정보를 이용한 수도-아웃라이어 제거 기법

지도 교수 박 찬 국

이 논문을 공학석사 학위논문으로 제출함
2019 년 12 월

서울대학교 대학원
기계항공공학부
정 재 영

정재영의 공학석사 학위논문을 인준함
2019 년 12 월

위 원 장 김 유 단 (인)

부위원장 박 찬 국 (인)

위 원 김 현 진 (인)

Abstract

Pseudo-Outlier Method in Vision-Aided Navigation System Using Prior Information

Jae Young Chung

Department of Mechanical and Aerospace Engineering

The Graduate School

Seoul National University

In this work, we proposed an algorithm designed for realistic vision-aided navigation systems. Random sample consensus, the most popularly used algorithm for vision-aided navigation systems, is not properly operated in real-world situations with pseudo-outliers like a moving object taking a large part of the image. The proposed method was designed to replace the conventional algorithm using prior information from an additional reliable sensor. We evaluated the proposed algorithm in a simulation to verify that it can filter out the large moving feature group like bus or truck and can achieve better performance than the conventional algorithm. We also applied our algorithm to a vision-aided wheel odometry system with a multi-state constraint Kalman filter in a real-time system. The results demonstrate that the proposed algorithm prevents the malfunction of the classic algorithm and improves the accuracy of position estimation, especially when there is a pseudo-outlier.

Keywords: Vision-aided navigation system, Pseudo-outlier, Multi-state constraint Kalman filter, Random sample consensus, Mean shift clustering,

Student Number: 2018-22641

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Motivation and background

Localization is widely used in the various areas on scientific to industrial purposes. Since the global navigation satellite system provides absolute position on Earth, one can easily know its own position with a small electronic module. However, there are many of harsh environment such as the indoor environment, the war field with gamming system and even the extraterrestrial planets. In these situations, another localization methods from the circumstance near the body are needed.

Visual information is the most popular method to interact with the circumstance through the ray of lights. There are active and passive method. For example, LiDAR, [32] RADAR, and Sonar sensors are active method because they emits some rays and counts the time that the reflected rays are arrived. In contrast, electro-optical camera and Infrared camera are passive method as they just receive the rays from the environment. Generally, the word *vision* means only the passive methods which is made with cameras. In this context, we also use the word as indicating the methods with cameras.

As the most of animals gets the most of the information of environment with their eyes, the vision system can highly effect on the performance of localization system. However, the navigation systems

based on the vision suffers from the low accuracy. In order to increase the accuracy of the navigation system, all the process from generating visual information to selecting good information must be treated carefully.

Random sample consensus [1] is one of the most popular method in vision system to distinguish a set of data. However, due to fundamental limitations, this method is hard to be applied in real-world localization solutions. As they assumes that the most of the data comes from the environment, they cannot operate properly when there is a large moving object that covers all of the view of the camera.

Furthermore, there were some accidents of autonomous vehicles reported that the vision system misjudges and leads to a fatal action. [34] The investigator reported that due to the white large trucks on each side of the vehicle, the car misjudged its speed and direction. As a result, we learned from these unfortunate accidents that we have to develop some method to divide moving objects and environments.

In these days, there are many of approaches with deep learning. From 2018, researchers in localization field have used the deep neural network as a tool to make it accurately. [33] In spite of their efforts on the field, it is hard to guarantee the safety action of the neural network. Furthermore, deep neural network is not appropriate to the small embedded systems in automobiles.

Therefore, we researched the way to improve the accuracy and safety of the vision-aided autonomous vehicles in classic indirect vision field.

In the same context, we adopt the odometry scheme instead of the simultaneous localization and mapping. (SLAM) [35]

1.2 Objectives and contributions

In order to overcome the disadvantage of random sample consensus assuming the largest group as an answer, this work proposed a method filtering out with reliable sensors. Not only are the additional reliable sensors good to drive the whole system, it can be a good source of ego-motion information to the vision system. Definitely, vision-aided navigation system on the automobile needs to be filtered out the moving cars to estimate ego-motion properly. The method proposed in this work can be one solution to develop the outdoor navigation solution. Finally, the last goal is to make a system which is not interrupted by moving objects in front of our automobile.

The proposed method can be applied to any vision-aided navigation system with some reliable sensor. In this work, we applied the method on a visual-inertial-wheel odometry system formed with multi-state constraint Kalman filter. (MSCKF) [2] Although the original paper of MSCKF is formed with inertial measurement unit and monocular vision system, we took the wheel odometry data due the fatal accuracy of low-cost accelerometer.

Chapter 2. Related Works

2.1 Vision-aided navigation system

Visual odometry is first adopted in the Mars explorer robot [3] to use at sandy circumstances. Although it needs to be record its position, there is no global navigation satellite system to know its absolute position on Mars. Therefore, the group of engineers adopt a relative position system called odometry. Wheel odometry is one of the most accurate relative positioning method that day, however, large slippery error occurs on Mars because there are only sands and rocks. Therefore, visual odometry is researched to apply on the explorer robot in [3, 4]. Visual information is combined with the other sensors nowadays, including inertial measurement unit (IMU) with accelerometer and gyroscope [5, 6], global navigation satellite system (GNSS) [7], even wheel odometry. When the sensors are reliable and have complementary properties compensating each other, the fused system can achieve a higher performance than each of the sensor.

Vision-aided navigation system is divided into some groups along the number of camera used to construct the system. Monocular vision means that the system uses only one camera in the system to estimate. Stereo vision means that there are two cameras to solve the given problem. Monocular visual odometry is better than stereo visual

odometry with respect to the low cost, the low computation time, and the low complexity to estimate the ego-motion. However, due to the scale ambiguity, monocular camera suffers to know exact navigation solution. If there is another sources of information of ego-motion, monocular system can take an advantage of low cost without scale ambiguity.

In fact, there is various ways to estimate the motion of camera with respect to the mapping and estimation method. Visual Simultaneous Localization and Mapping, or Visual SLAM is a method to get its position and the information of environment near the camera with the vision sensor. For example, we can construct the feature map with the indirect visual information like point features. Although it needs many of memories to save the feature maps, it can compensate its position when there is a closed loop on the trajectory. That is, when the camera goes to the same way, the system can specify where it is on the feature map, used the estimated position and position on feature map to compensate its position. This is widely used in indoor navigation system because of the memory limitation on the outdoor navigation. In contrast, visual odometry does not save any information of the environment driving local positioning between the previous frame and current frame as shown in Figure 2.1. Since this system does not have any method to correct its position, the position error is getting bigger as navigation time goes on. In outdoor navigation, however, there is a rare chance of driving through the same way and the limitation of the computing sources, visual SLAM was not a proper method for the system on the automobile.

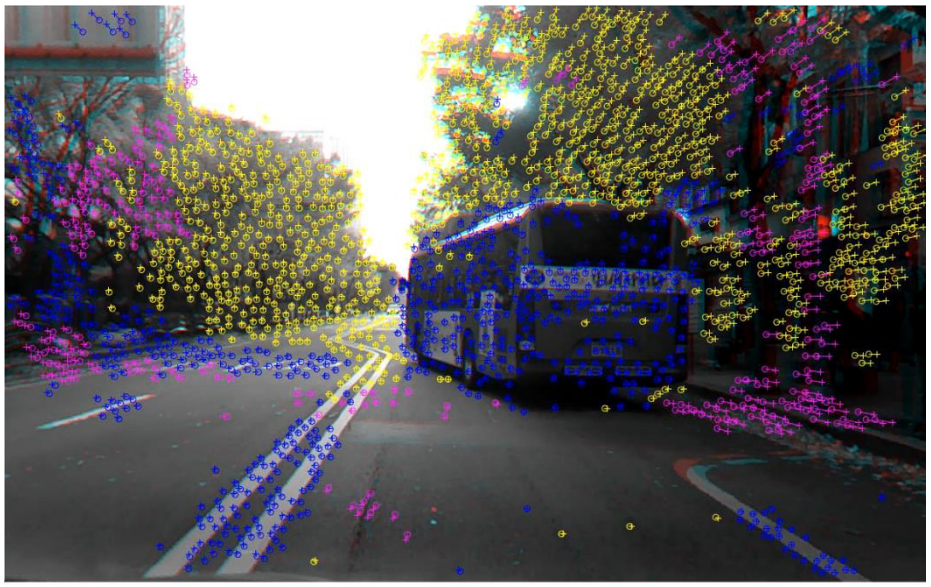


Figure 2.1. An example image in odometry with tracked features

There are two approaches in vision-aided navigation system. One is optimization method and the other is filtering method. Optimization method focus on minimizing the cost between relative frames. For example, [8] focus on the Epipolar geometry [36] on two view and optimize the multiplication of translation vector and rotation matrix. We can finally find each element with a mathematical constraint. Note that the optimization scheme is not limited on the two view geometry. In case of filtering based method, there is a system propagation model and measurements to update the state of the system. In visual odometry system, since there is no other sensor to drive the system, generally assumes the system as constant velocity model in short time. In vision-aided navigation system, there are additional sensors to drive the system.

Inertial measurement unit is one of the sensors that drive a navigation system. This system is called visual-inertial odometry. Furthermore, wheel odometry is also a candidate sensor to drive a navigation system. Although visual odometry is researched to overcome the disadvantage of the wheel odometry in a harsh environment, they can be combined in certain situation to increase the performance of navigation system.

2.2 Random sample consensus

Many of vision-aided navigation systems are adopt methods to select proper features. Random sample consensus, called RANSAC from RANdom SAmple Consensus, is the most preferred method in the indirect vision systems. This method is an iterative method finding the parameters of a given hypothesis from the measured data. [1] While it repeat the hypothesis test from randomly selected data, the iteration number is determined from the probability of inlier ratio to get the most supportive parameter. The most supportive hypothesis means a hypothesis describing all the data well. With an arbitrary generated hypothesis, the cost function is defined with a threshold or mean squared error. In many of papers in vision system RANSAC is preferred because it is easy to use without any complex optimization, only needs a little amounts of memory.

The details of the method follows. First of all, we assume that we have a set of observed data and the knowledge of the mathematical

structure of data. After choosing a minimum subset of the data randomly, generates a hypothesis from the subset. This hypothesis has candidate parameters of this time step. With a cost function such as a threshold or mean squared error, calculates the cost of each data points for the hypothesis of this time step, and distinguishes the data into inlier and outlier with the criterion. The reason of calculating the cost function is that we finally select the hypothesis having the smallest cost as the answer because it has the largest subset of data. Therefore, from the iterative process, we choose a hypothesis which is supported from the highest number of data points. The theory of random sample consensus proposed the number of iteration to achieve certain level of success probability.

$$t = \frac{\log(1-\alpha)}{\log(1-\gamma^m)} \tag{2.1}$$

where t is the number of iteration to achieve the given probability to success α in a set of data having inlier ratio γ that needed m number of data to generate the hypothesis. In fact, the inlier ratio γ is unknown variable because we don't know whether each point is inlier or not. Therefore, we approximately estimated the inlier ratio and assign a value under the estimated inlier ratio to γ. With this reason, tutorial [9] recommends the number of iterations more than three times of $\boldsymbol{t}$.

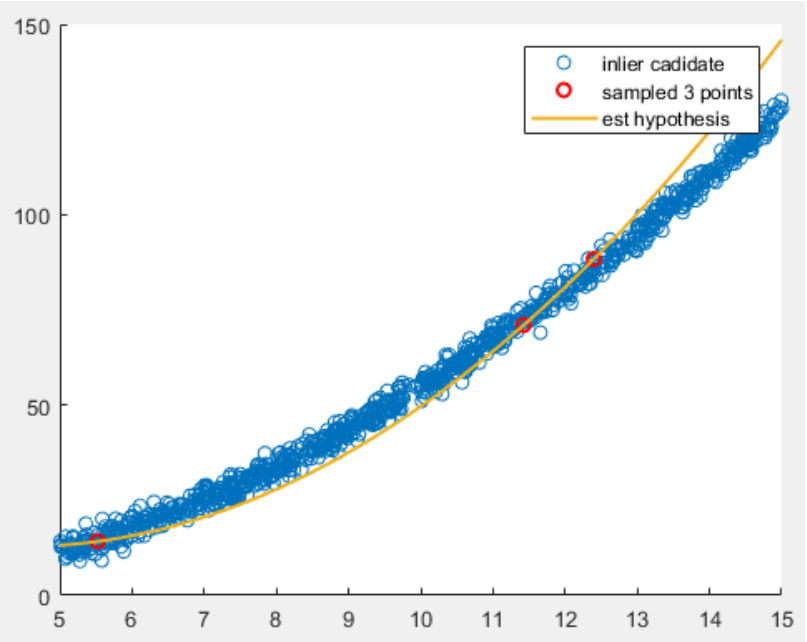


Figure 2.2. A toy example of random sample consensus

We can understand the detailed algorithm of random sample consensus with a toy problem as follows. Figure 2.2. describes an example usage of RANSAC with data that follows the parabola. We know the structure that the data follows, except the exact parameters of parabola. From the knowledge that 3 points are needed to construct a parabola, we randomly select 3-points from the set of data. Figure 2.3. shows the estimated hypothesis from the randomly selected data points and the division of inliers (blue points) and outliers. (black points) We selected inliers within a threshold, and cost function of each hypothesis is designed as the ratio of inliers to all the data points. If the hypothesis made in this time step is the most supportive hypothesis, or it contains

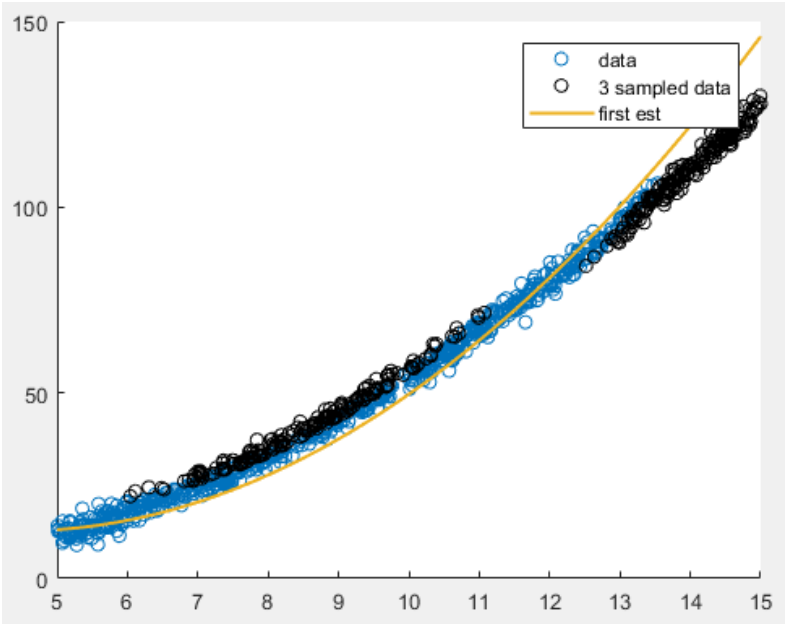


Figure 2.3. An example hypotheis during the iteration of RANSAC

the highest number of data points, we saved the hypothesis and inliers temporarily. We repeated this process with the success probability $\alpha = 0.99$, $\mathrm{m} = 3$ points for parabola. We assumed the inlier ratio as $\gamma = 0.5$. Therefore, we set the iteration number $\boldsymbol{t}$ to 104 as described in (2.2).

$$t = 3 \cdot \frac{\log(1 - 0.99)}{\log(1 - 0.5^3)} \approx 104 \qquad (2.2)$$

The idea of random sample consensus is widely used because it can reject the outliers without any prior knowledge of the true value. In fact, the method treat the largest group with the same tendency as the answer of the parameter finding problem with a given mathematical model. It showed powerful performances on some specific problems such as model fitting problem, Epipolar geometry estimation, motion estimation,

segmentation, structure from motion, feature-based localization.

There are many of upgrade versions of random sample consensus with respect to the accuracy, speed and robustness. At first, new lose functions, local optimization methods and model selection methods are proposed to improve the accuracy of random sample consensus. For example, as redesigning the loss function that is used to choose the best hypothesis, they intend to lower the boundary across the inlier and outlier. MLESAC [10] assume that the group of inlier has Gaussian distribution and the outliers have uniform distribution. MAPSAC [11] also takes Bayesian approach with the loss function similar with the one of MSAC. [10]

Next, guided sampling methods and partial evaluation is designed to increase the speed of the method. They focus on decreasing the number of the evaluation as giving some hint instead of selecting randomly, and evaluating each hypothesis with the subset of data points. For example, Guided MLESAC [12] have a probability distribution instead of the uniform distribution as prior knowledge, and NAPSAC [13] uses N adjacent points to sample. Also, instead of evaluating all the data points, evaluating partially decrease the time to test each hypothesis to choose the best parameters. Randomized RANSAC [14] is one of the partial evaluation method as testing each hypothesis with a random subset of data points.

Last, there are some adaptive methods for robustness. Adaptive evaluation is designed to select adaptively the threshold because the

value is highly dependent on the data. For example AMLESAC [15] uses gradient descent and expectation and maximization (EM) method [16] to search a proper threshold. Adaptive termination intend to increase the robustness and speed. After the first evaluation, we can observe that the inlier ratio is monotonically increased because we choose a hypothesis that contains data points as many as possible.

More details are described in [17]. Although there are many work to improve the performance of random sample consensus with respect to the accuracy, speed and robustness, there is fundamental problems on the methods based on the RASAC. The first problem is that the methods on random sample consensus assume that there is only one group on data. Even if there are more than one group on data, there regard all the data points as outlier except the data points in largest inlier group. The second problem is that they assume the largest group as the answer. This scheme is derived from the situation that there is no other source of information to choose properly. In visual odometry case, the camera is the only one to get information about its ego-motion. Therefore the methods based on random sample consensus is inevitably recommend to increase the accuracy of estimation.

In [18], *pseudo-outlier* is first introduced meaning data points that is inlier to one group but also outlier to the other groups. It intends to consider a data with many groups. The basic method is operating the random sample consensus sequentially. Since this method have a low performance, multiRANSAC [19] is proposed as grouping

simultaneously, not sequentially. Linkage methods such as J-linkage [20] and T-linkage [21] take another approach opposite to the way of sample consensus. Instead of generating a hypothesis from the subset of data points, linkage scheme has a number of hypothesis to test each data point. Calculating the probability that a data point belongs to each hypothesis. With this approach, we can find multiple groups in ascending order of the number of data points. Nonparametric estimation of multiple structure such as kernel density estimation (KDE) [22] and k-nearest neighbor density estimation [22] are also useful methods to clustering the given data. Mean shift [23, 28] is a powerful method to find the local centers where the data points are densely distributed. There are also optimization methods such as PEaRL [24].

Even though these methods can be applied to divide obstacles on the image, they cannot judge whether the data is proper to estimate the ego-motion of camera. The essential problem is arisen when there is a moving obstacle occupying the large portion of the image. For example, Figure 2.4. shows the situation that a large truck goes across in front of the camera. Even though the camera is in the stationary situation, the ego-motion estimation with random sample consensus judge that the camera goes to the right. Due to this reason, many of papers used datasets such as KITTI [25] and EuroC [26] to ignore this problem. This can be shown as a small problem to make vision-aided navigation system. However, in order to use the navigation solution in real-world environment, this is an inevitable essential problem must be solved.

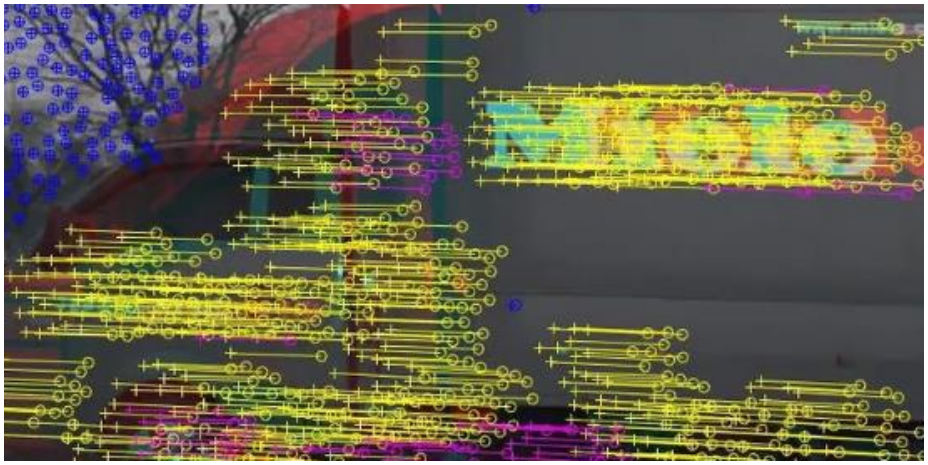


Figure 2.4. The result of RANSAC when a truck goes across the view

In the vision-aided navigation system including other sensors except the camera, they can be a guideline to the vision sensor. This is the key idea of the proposed method. We applied this idea to vision-aided navigation system with an inertial measurement unit and wheel odometry using modified multi-state constraint Kalman filter.

Chapter 3. Multi-state Constraint Kalman Filter for Mono Visual-inertial-wheel Odometry

We choose multi-state constraint Kalman filter as the driving system of visual-aided system. This system is classified EKF-based, odometry, monocular method with a specific feature treatment. The original paper [2] constructed the system only with inertial measurement unit and a monocular vision, our system select a low cost MEMS gyroscope and wheel odometry with controller area network (CAN) communication system in an automobile. In this chapter, we first introduce the original MSCKF with respect to the system update, and we finally describe the modified MSCKF maintaining the key idea of it.

The notations in this work determined as follows. A vector is represented by a small bold letter, and a matrix is represented by a capital normal letter. A vector $\boldsymbol{p}_{GB}^{G}$ represents a position vector from global frame {G} to body frame {B} with respect to global frame {G}. A matrix R_{B}^{G} represents a rotation matrix included in $\mathrm{SO}(3)$. This matrix R_{B}^{G} transforms the frame from {B} to {G} as $\boldsymbol{p}^{G} = \mathrm{R}_{B}^{G}\boldsymbol{p}^{B}$. For an arbitrary vector $\boldsymbol{w}$, $[\boldsymbol{w}\times]$ is a skew- or anti-symmetric matrix of $\boldsymbol{w}$ such that

$$[\boldsymbol{w}\times] = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \tag{3.1}$$

Furthermore, vectors with a hat infer that it is an estimated vector, and

vectors with tilde mean that it is an error state vector which is defined as

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}} \tag{3.2}$$

Note that the vector without any hat is the vector of true value.

3.1 Multi-state constraint Kalman filter

First of all, in order to clarify the discussion, we notice the notations and frames. We define global frame {G}, body frame {B}, camera frame {C}, IMU sensor frame {S} and inertial frame {I}. Each frame is defined in order to describe the data of each sensor, global frame is defined in the range of the navigation system. The transform matrix $\mathrm{T} \in \mathrm{SE}(3)$ is also considered as defining directly or estimating the relations of frames within the filter.

Although the global frame can be defined as Earth-Centered Earth-Fixed (ECEF) frame, we defined the global frame with local tangent plane and the gravity. This frame is occasionally called '*local frame*', in order to avoid the confusion between the word *global* and *local*, we just call the local tangent plane and gravity as *global* frame. The axes of x, y and z directs to the north, east and downward directions, respectively. The definition of global frame is totally dependent on the usage of the system. If the system was designed to use in terrestrial level, the local tangent plane with the gravity vector is appropriate. However, if the system was designed to orbit on Earth or more, one should use ECEF or Sun-centered frame.

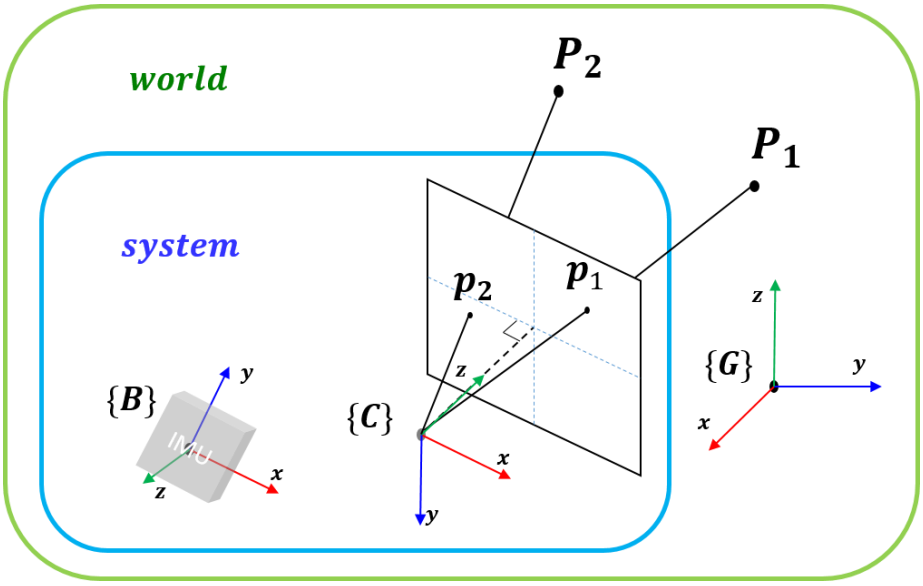


Figure 3.1. The original system configuration of the MSCKF

The body frame {B} is coincided with the IMU sensor frame {S} in the original MSCKF. Since there is no other constraint with the position of body frame, the original paper easily assign the IMU frame to the body frame {B}. Therefore, we consider the frames {G}, {B}≡{S}, {C} to described the motion of system for navigation. We also consider only the transformation between {G} and {B} as T_B^G, and the transformation between {B} and {C} as T_C^B.

We select the camera frame {C} as the normalized camera frame with the focal length 1. The origin of normalized camera frame is located at the point that is 1 away from the image plane. The projected point of the origin is the center of the image plane. The axes x, y of the normalized image plane is aligned with the horizontal and vertical axes of the image plane and the z axes is perpendicular with the image plane.

Overall the definitions and relations are described in Figure 3.1.

3.1.1 State representation

We defined the IMU state and the error state as (3.3).

$$\mathbf{x}_{\mathrm{IMU}} = \begin{bmatrix} \bar{\mathbf{q}}_{\mathrm{B}}^{\mathrm{G}} \\ \mathbf{p}_{\mathrm{GB}}^{\mathrm{G}} \\ \mathbf{v}_{\mathrm{B}}^{\mathrm{G}} \\ \mathbf{b}_{\mathrm{a}} \\ \mathbf{b}_{\mathrm{g}} \end{bmatrix}, \qquad \tilde{\mathbf{x}}_{\mathrm{IMU}} = \begin{bmatrix} \boldsymbol{\delta\theta}_{\mathrm{B}}^{\mathrm{G}} \\ \tilde{\mathbf{p}}_{\mathrm{GB}}^{\mathrm{G}} \\ \tilde{\mathbf{v}}_{\mathrm{B}}^{\mathrm{G}} \\ \tilde{\mathbf{b}}_{\mathrm{a}} \\ \tilde{\mathbf{b}}_{\mathrm{g}} \end{bmatrix} \tag{3.3}$$

where $\bar{\mathbf{q}}_{\mathrm{B}}^{\mathrm{G}}$, $\mathbf{p}_{\mathrm{GB}}^{\mathrm{G}}$, $\mathbf{v}_{\mathrm{B}}^{\mathrm{G}}$ are the unit quaternion of rotation, the position vector, and velocity vector from global frame {G} and to body frame {B}, respectively. The last two terms b_{a} and b_{g} are the biases of accelerometer and gyroscope. As a result, the IMU state is constructed with 16 dimensions. The error state is a little bit different with its original state. Since the error quaternion $\delta\bar{\mathbf{q}}$ which is defined as $\bar{\mathbf{q}} = \delta\bar{\mathbf{q}} \otimes \hat{\bar{\mathbf{q}}}$ can be approximated as (3.4)

$$\delta\bar{\mathbf{q}} \backsimeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta} \\ 1 \end{bmatrix} \tag{3.4}$$

Although we can use the error quaternion, consisting with the small rotation is profit to decrease the order of the state. That is the minimum representation of the rotation error.

MSCKF suggests to augment the pose of the last N camera frames, while conventional EKF-based methods keep track the 3d position of the

features. At current time step k, the total state vector containing the last N camera frames is described as (3.5)

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{x}}_{\mathrm{IMU}_k} \\ \boldsymbol{q}_{C_{k-N}}^{G} \\ \boldsymbol{p}_{C_{k-N}}^{G} \\ \vdots \\ \boldsymbol{q}_{C_{k-1}}^{G} \\ \boldsymbol{p}_{C_{k-1}}^{G} \end{bmatrix}, \qquad \tilde{\mathbf{x}}_{\mathrm{k}} = \begin{bmatrix} \tilde{\mathbf{x}}_{\mathrm{IMU}_k} \\ \tilde{\boldsymbol{q}}_{C_{k-N}}^{G} \\ \tilde{\boldsymbol{p}}_{C_{k-N}}^{G} \\ \vdots \\ \tilde{\boldsymbol{q}}_{C_{k-1}}^{G} \\ \tilde{\boldsymbol{p}}_{C_{k-1}}^{G} \end{bmatrix} \tag{3.5}$$

All the camera frames are described in global frame. With this state representations, the system propagation and measurement model is constructed, including the Jacobian matrix and the special processing for the measure.

3.1.2 System model and update

The system in continuous time is described with respect to the derivative of the state. The quaternion for body attitude is derived from the gyroscope data. The position and velocity is derived from the accelerometer data.

$$\begin{bmatrix} \dot{\mathbf{q}}_{\mathrm{B}}^{\mathrm{G}} \\ \dot{\mathbf{p}}_{\mathrm{GB}}^{\mathrm{G}} \\ \dot{\mathbf{v}}_{\mathrm{B}}^{\mathrm{G}} \\ \dot{\mathbf{b}}_{\mathrm{a}} \\ \dot{\mathbf{b}}_{\mathrm{g}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Omega\left(\mathbf{w}_{\mathrm{m}} - \mathbf{b}_{\mathrm{g}} - \mathbf{n}_{\mathrm{g}}\right)\bar{\mathbf{q}}_{\mathrm{B}}^{\mathrm{G}} \\ \mathbf{v}_{\mathrm{B}}^{\mathrm{G}} \\ \mathrm{R}_{\mathrm{B}}^{\mathrm{G}}(\mathbf{a}_{\mathrm{m}} - \mathbf{b}_{\mathrm{a}} - \mathbf{n}_{a}) + \mathbf{g}^{G} \\ \mathbf{n}_{\mathrm{ba}}(t) \\ \mathbf{n}_{\mathrm{bg}}(t) \end{bmatrix} \tag{3.6}$$

where the vectors $\boldsymbol{w}_m(\mathrm{t})$ and $\boldsymbol{a}_m(\mathrm{t})$ are the data sequence along the

time, $\mathbf{n}_{\mathrm{ba}}(t)$ and $\mathbf{n}_{\mathrm{bg}}(t)$ are Gaussian random process. $\Omega(\boldsymbol{w})$ is defined as (3.7)

$$\Omega(\boldsymbol{w}) = \begin{bmatrix} -[\boldsymbol{w}\times] & \boldsymbol{w} \\ -\boldsymbol{w}^T & 0 \end{bmatrix} \tag{3.7}$$

The data from gyroscope and accelerometer is modeled with bias and Gaussian noise. Therefore, the measurement of these value from each sensor is modeled as (3.8)

$$\begin{aligned} \mathbf{w} &= \mathbf{w}_{\mathrm{m}} - \mathbf{b}_{\mathrm{g}} - \mathbf{n}_{\mathrm{g}} \\ \mathbf{a}^{\mathrm{G}} &= \mathrm{R}_{\mathrm{B}}^{\mathrm{G}}(\mathbf{a}_{\mathrm{m}} - \mathbf{b}_{\mathrm{a}} - \mathbf{n}_{a}) + \mathbf{g}^{G} \end{aligned} \tag{3.8}$$

Since we defined the global frame as local tangent plane on Earth, there are no complicated terms of the effect from the motion of Earth.

After estimating the model of this system, we can finally define the error model as state-space form. The estimation form of the state is derived from the expectation of the original state as (3.9).

$$\begin{bmatrix} \dot{\hat{\mathbf{q}}}_{\mathrm{B}}^{\mathrm{G}} \\ \dot{\hat{\mathbf{p}}}_{\mathrm{GB}}^{\mathrm{G}} \\ \dot{\hat{\mathbf{v}}}_{\mathrm{B}}^{\mathrm{G}} \\ \dot{\hat{\mathbf{b}}}_{\mathrm{a}} \\ \dot{\hat{\mathbf{b}}}_{\mathrm{g}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\widehat{\Omega}\left(\mathbf{w}_{\mathrm{m}} - \hat{\mathbf{b}}_{\mathrm{g}}\right)\widehat{\mathbf{q}}_{\mathrm{B}}^{\mathrm{G}} \\ \hat{\mathbf{v}}_{\mathrm{B}}^{\mathrm{G}} \\ \widehat{\mathrm{R}}_{\mathrm{B}}^{\mathrm{G}}\left(\mathbf{a}_{\mathrm{m}} - \hat{\mathbf{b}}_{\mathrm{a}}\right) + \hat{\mathbf{g}}^{\boldsymbol{G}} \\ \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \end{bmatrix} \tag{3.9}$$

With the simple definition of error state $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$, the system model is totalized at (3.11)

$$\dot{\tilde{\mathbf{x}}}_{\mathrm{IMU}} = \begin{bmatrix} -[\hat{\mathbf{w}}\times]\delta\boldsymbol{\theta} - \mathbf{b}_{\mathrm{g}} - \mathbf{n}_{\mathrm{g}} \\ -\mathrm{R}_{\mathrm{B}}^{\mathrm{G}}([\hat{\boldsymbol{a}}\times]\delta\boldsymbol{\theta} + \mathbf{b}_{\mathbf{a}} + \mathbf{n}_{\mathbf{a}}) \\ \tilde{\mathbf{v}}_{\mathrm{B}}^{\mathrm{G}} \\ \mathbf{n}_{\mathrm{ba}} \\ \mathbf{n}_{\mathrm{bg}} \end{bmatrix} \tag{3.10}$$

$$= \begin{bmatrix} -[\hat{\mathbf{w}}\times] & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\mathrm{I}_3 \\ -\mathrm{R}_{\mathrm{B}}^{\mathrm{G}}[\hat{\boldsymbol{a}}\times] & 0_{3\times3} & 0_{3\times3} & -\mathrm{R}_{\mathrm{B}}^{\mathrm{G}} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \mathrm{I}_3 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tilde{\mathbf{x}}_{\mathrm{IMU}} + \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & -\mathrm{I}_3 & 0_{3\times3} \\ -\mathrm{R}_{\mathrm{B}}^{\mathrm{G}} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \mathrm{I}_3 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \mathrm{I}_3 \end{bmatrix} \mathbf{n}_{\mathrm{IMU}} \tag{3.11}$$

$$= \mathrm{F}\tilde{\mathbf{x}}_{\mathrm{IMU}} + \mathrm{G}\mathbf{n}_{\mathrm{IMU}} \tag{3.12}$$

where $\mathbf{n}_{\mathrm{IMU}} = \begin{bmatrix} \boldsymbol{n}_a^T & \boldsymbol{n}_{ba}^T & \boldsymbol{n}_g^T & \boldsymbol{n}_{bg}^T \end{bmatrix}^T$. The effect of the rotating Earth is ignored since we see the local tangent plane on Earth surface as global frame.

(3.13) is the system propagation model on the discrete state space.

$$\mathbf{x}_k^- = \boldsymbol{\phi}_{\mathrm{k-1}}\mathbf{x}_{\mathrm{k-1}}^+ \tag{3.13}$$

With a carefully selected numerical integration method, for example 5th order Runge-Kutta numerical integration, we integrate the state values and data sequence from IMU. Furthermore, we have to propagate the covariance matrix to determine the estimation boundary as (3.14).

$$\text{For } \mathrm{P}_{k-1}^{+} = \begin{bmatrix} \mathrm{P_B} & \mathrm{P_{BC}} \\ \mathrm{P_{CB}} & \mathrm{P_C} \end{bmatrix}, \mathrm{P}_k^{-}$$

$$= \begin{bmatrix} \boldsymbol{\phi}_{k-1}\mathrm{P_B}\boldsymbol{\phi}_{k-1}^{\mathrm{T}} + \mathrm{Q_{IMU}} & \boldsymbol{\phi}_{k-1}\mathrm{P_{BC}^{-}} \\ \mathrm{P_{CB}^{-}}\boldsymbol{\phi}_{k-1}^{\boldsymbol{T}} & \mathrm{P_C^{-}} \end{bmatrix} \tag{3.14}$$

where $\boldsymbol{\phi}_{\mathbf{k-1}}$ is the transition matrix of the system. The transition matrix and Lyapunov equation are formed as (3.15) and (3.16)

$$\dot{\boldsymbol{\phi}}(\tau, t_k) = \mathrm{F} \cdot \boldsymbol{\phi}(\tau, t_k), \qquad \tau \in [t_k, t_{k+1}] \tag{3.15}$$

$$\dot{\mathrm{P}}_{\mathrm{B}} = \mathrm{F}\mathrm{P}_{\mathrm{B}} + \mathrm{P}_{\mathrm{B}}\mathrm{F}^{\mathrm{T}} + \mathrm{G}\mathrm{Q}_{\mathrm{IMU}}\mathrm{G}^{\mathrm{T}} \tag{3.16}$$

where $\mathrm{Q_{IMU}}$ is the power spectral density of the IMU noise in continuous domain.

When a new image is recorded, the camera 6 DOF pose is calculated from the IMU data in global frame. This fresh two vectors are augmented in the state, and the oldest camera pose is replaced. Therefore, there are up-to-date N numbers of camera poses on the state. Moreover, the covariance matrix also have to be changed as (3.17)

$$\mathrm{P}^{-} \leftarrow \begin{bmatrix} P^{-} & \mathrm{P}^{-}\boldsymbol{\mathcal{J}}_{\mathrm{IMU}}^{\mathrm{T}} \\ \boldsymbol{\mathcal{J}}_{\mathrm{IMU}}\mathrm{P}^{-} & \boldsymbol{\mathcal{J}}_{\mathrm{IMU}}\mathrm{P}^{-}\boldsymbol{\mathcal{J}}_{\mathrm{IMU}}^{\mathrm{T}} \end{bmatrix} \tag{3.17}$$

$$\boldsymbol{\mathcal{J}}_{\mathrm{IMU}} = \begin{bmatrix} R_C^B & 0_{3\times9} & 0_{3\times3} & 0_{3\times6N} \\ [R_B^G p_{BC}^B \times] & 0_{3\times9} & \mathrm{I}_{3\times3} & 0_{3\times6N} \end{bmatrix}$$

where the matrix $\boldsymbol{\mathcal{J}}_{\mathrm{IMU}}$ is the Jacobian matrix derived from the IMU propagation equation. This process is repeated every new camera frames are measured.

3.1.3 Measurement model and update

The measurement is construct from the data, especially indirect point features. The measurement residual $\mathbf{r}$ is defined with the position of the observed features on the image plane. Assuming that there are observed features f_j at the current image. Features are tracked during a few frames or newly detected at the current frame. MSCKF uses the stacked features S_j as the materials to make its measurement vector. In each stacked features, there are the observed points on each camera. For the feature f_j, we represents the i^{th} features at the i^{th} camera frame as (3.18).

$$\mathbf{z}_i^{(\mathrm{j})} = \frac{1}{Z_j^{C_i}} \begin{bmatrix} X_j^{C_i} \\ Y_j^{C_i} \end{bmatrix} + \mathbf{n}_i^{(\mathrm{j})}, \qquad \text{for } i \in S_j \tag{3.18}$$

As defining the error state $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$, we calculate the measurement residual as (3.19).

$$\mathbf{r}_i^{(\mathrm{j})} = \mathbf{z}_i^{(\mathrm{j})} - \hat{\mathbf{z}}_i^{(\mathrm{j})} \tag{3.19}$$

The works on MSCKF suggests an approximation of the residual with the feature position in global frame.

$$\text{From } \mathbf{p}_{f_j}^{C_i} = \begin{bmatrix} X_j^{C_i} \\ Y_j^{C_i} \\ Z_j^{C_i} \end{bmatrix} = \mathrm{R}_G^{\mathrm{C_i}} \left(\mathbf{p}_{f_j}^G - \mathbf{p}_{C_i}^G \right),$$

$$\mathbf{r}_i^{(\mathrm{j})} \simeq \mathrm{H}_{\mathbf{x}_i}^{(\mathrm{j})} \tilde{\mathbf{x}}^- + \mathrm{H}_{f_j}^{(\mathrm{j})} \tilde{\mathbf{p}}_{f_j} + \mathbf{n}_i^{(\mathrm{j})} \tag{3.20}$$

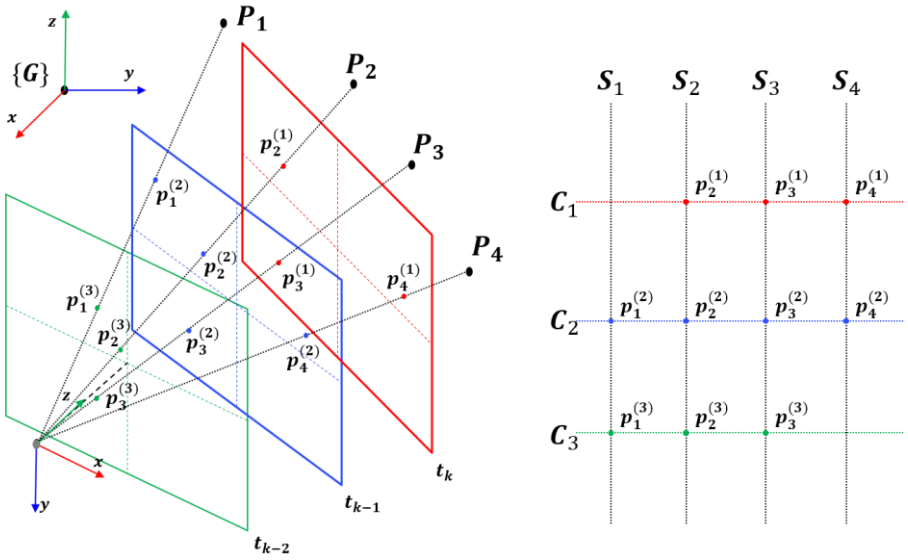


Figure 3.2. Observed features on multiple frames and stacked features

(3.20) is the carefully designed approximation where the measurement matrixes $\mathrm{H}_{\mathbf{x}_i}^{(\mathrm{j})}$ and $\mathrm{H}_{f_j}^{(\mathrm{j})}$ are defined with the Jacobians as (3.21), (3.22).

$$\mathrm{H}_{\mathbf{x}_i}^{(\mathrm{j})} = \begin{bmatrix} \mathbf{0}_{2\times 15} & \cdots & \boldsymbol{\mathcal{J}}_{\boldsymbol{i}}^{(\mathrm{j})} \left[\hat{\mathbf{x}}_{\boldsymbol{f_j}}^{\boldsymbol{C_i}} \times\right] & -\boldsymbol{\mathcal{J}}_{\boldsymbol{i}}^{(\mathrm{j})} \mathrm{R}_{\boldsymbol{G}}^{\boldsymbol{C_i}} & \cdots \end{bmatrix} \tag{3.21}$$

$$\mathrm{H}_{f_j}^{(\mathrm{j})} = \boldsymbol{\mathcal{J}}_{\boldsymbol{i}}^{(\mathrm{j})} \mathrm{R}_{\boldsymbol{G}}^{\boldsymbol{C_i}} \tag{3.22}$$

$$\text{where } \boldsymbol{\mathcal{J}}_{\boldsymbol{i}}^{(\mathrm{j})} = \frac{\partial \mathbf{z}_i^{(\mathrm{j})}}{\partial \hat{\mathbf{p}}_{f_j}^{C_i}} = \frac{1}{Z_j^{C_i}} \begin{bmatrix} 1 & 0 & -\frac{X_j^{C_i}}{Z_j^{C_i}} \\ 0 & 1 & -\frac{Y_j^{C_i}}{Z_j^{C_i}} \end{bmatrix} \tag{3.23}$$

Note that the Jacobian of IMU state $\boldsymbol{\mathcal{J}}_{\mathrm{IMU}}$ must be distinguished with the Jacobians of measurements $\boldsymbol{\mathcal{J}}_{\boldsymbol{i}}^{(\mathrm{j})}$. The detailed description of the stacked features are shown in Figure 3.2.

For each stacked features S_j, we use the stacked features as a measurement when the tracking ends. If the system missed the feature or the stack S_j were full, the stack S_j of feature f_j is now ready to be used as a measurement. Therefore, the stacked residual is represented at (3.24).

$$\mathbf{r}^{(\mathrm{j})} \simeq \mathrm{H}_{\mathbf{x}}^{(\mathrm{j})}\tilde{\mathbf{x}}^{-} + \mathrm{H}_{f_j}^{(\mathrm{j})}\tilde{\mathbf{p}}_{f_j} + \mathbf{n}^{(\mathrm{j})} \tag{3.24}$$

Since the naïve measurement residual is dependent on the position of features, it needs to be eliminate the term of feature position. Applying the null space of the $\mathrm{H}_{f_j}^{(\mathrm{j})}$ on both sides, MSCKF uses the modified measurement residual as (3.24). For the nullspace matrix $\mathrm{A} = \mathrm{null}\left(\mathrm{H}_{f_j}^{(\mathrm{j})}\right)$,

$$\mathbf{r}_O^{(\mathrm{j})} = \mathrm{A}^{\mathrm{T}}\mathbf{r}^{(\mathrm{j})} \simeq \mathrm{A}^{\mathrm{T}}\mathrm{H}_{\mathbf{x}}^{(\mathrm{j})}\tilde{\mathbf{x}}^{-} + \mathrm{A}^{\mathrm{T}}\mathrm{H}_{f_j}^{(\mathrm{j})}\tilde{\mathbf{p}}_{f_j} + \mathrm{A}^{\mathrm{T}}\mathbf{n}^{(\mathrm{j})} \tag{3.24}$$

$$= \mathrm{H}_O^{(\mathrm{j})}\tilde{\mathbf{x}}^{-} + \mathbf{n}_O^{(\mathrm{j})}$$

After preparing the measurement residuals to update, they added a simple update scheme with QR decomposition. Assembling all the features to update, they construct a totalized residual vector $\mathbf{r}_O$ as

$$\mathbf{r}_O = \mathrm{H}_O\tilde{\mathbf{x}}^{-} + \mathbf{n}_O \tag{3.25}$$

using QR decomposition on H_O, (3.6)is composed due to the orthogonality.

$$\mathrm{H}_O = [\mathrm{Q}_1 \quad \mathrm{Q}_2]\begin{bmatrix}\mathrm{T_H}\\ 0\end{bmatrix}$$

$$\mathbf{r}_O = [\mathrm{Q}_1 \quad \mathrm{Q}_2]\begin{bmatrix}\mathrm{T_H}\\ 0\end{bmatrix}\tilde{\mathbf{x}}^- + \mathbf{n}_O$$

$$\begin{bmatrix}\mathrm{Q}_1^T\mathbf{r}_O\\ \mathrm{Q}_2^T\mathbf{r}_O\end{bmatrix} = \mathbf{r}_O = \begin{bmatrix}\mathrm{T_H}\\ 0\end{bmatrix}\tilde{\mathbf{x}}^- + \begin{bmatrix}\mathrm{Q}_1^T\mathbf{n}_O\\ \mathrm{Q}_2^T\mathbf{n}_O\end{bmatrix} \tag{3.26}$$

The second equation with Q_2 is identity, we finally obtain the residual. Therefore, the rest equations is (3.27)

$$\mathbf{r}_\mathrm{n} = \mathrm{Q}_1^T\mathbf{r}_O = \mathrm{T_H}\tilde{\mathbf{x}}^- + \mathrm{Q}_1^T\mathbf{n}_O = \mathrm{T_H}\tilde{\mathbf{x}}^- + \mathbf{n}_n \tag{3.27}$$

In order to update the filter, (3.28) shows the Kalman gain of EKF.

$$\mathrm{K_k} = \mathrm{P_k^-}\mathrm{T_H^T}\left(\mathrm{T_H}\mathrm{P_k^-}\mathrm{T_H^T} + \mathrm{R}_{n,k}\right)^{-1} \tag{3.28}$$

The next state is calculated as adding the previous state and error state like in (3.29).

$$\mathbf{x}_\mathrm{k}^+ = \mathbf{x}_\mathrm{k}^- + \Delta\mathbf{x} = \mathbf{x}_\mathrm{k}^- + \mathrm{K_k}\mathbf{r}_\mathrm{n,k} \tag{3.29}$$

The covariance matrix is determined as (3.30)

$$\begin{aligned}\mathrm{P}_k^+ &= \left(\mathrm{I} - \mathrm{K_k}\mathrm{T}_{\mathrm{H}k}\right)\mathrm{P}_k^-\left(\mathrm{I} - \mathrm{K_k}\mathrm{T}_{Hk}\right)^T + \mathrm{K_k}\mathrm{R}_{n,k}\mathrm{K_k^T}\\ &= \left(\mathrm{I} - \mathrm{K_k}\mathrm{T}_{Hk}\right)\mathrm{P}_k^-\end{aligned} \tag{3.30}$$

3.2 MSCKF with wheel odometry

As we apply the MSCKF framework to our system, the body frame and the state vector is redefined. The center of body frame {B} is coincident with the midpoint of a non-steered axle where non-holonomic constraint (NHC) is ideally valid. The x-y-z axes of {B} points forward, right and

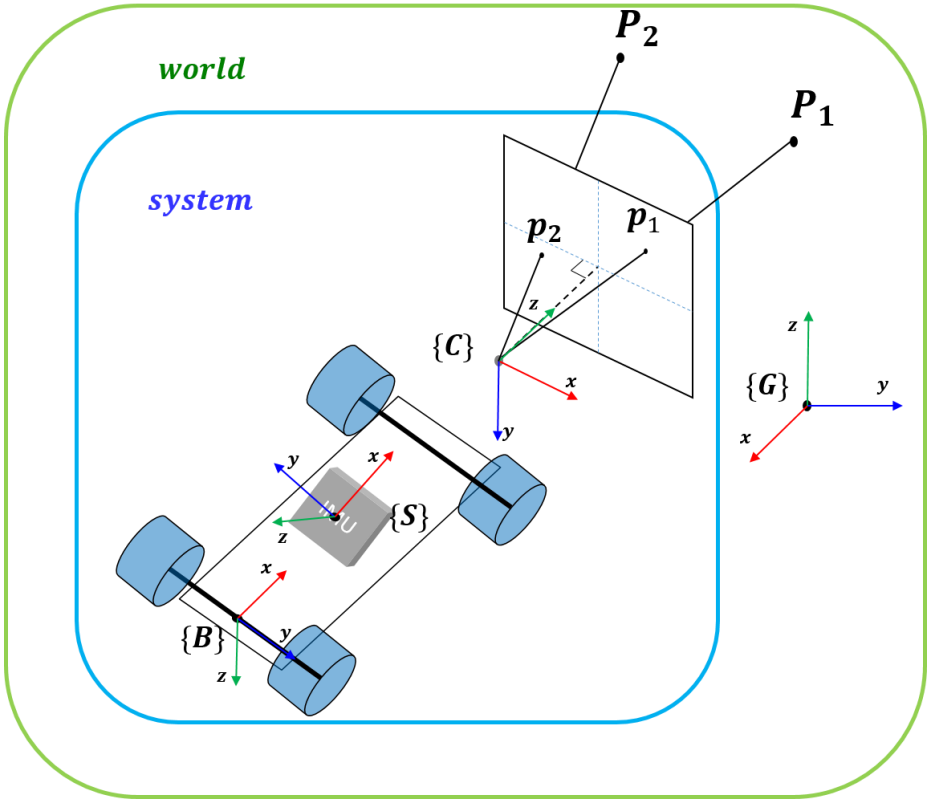


Figure 3.3. The system configuration of our modified MSCKF

down, respectively. Although the body frame and the IMU sensor frame {S} is different, we aligned the gyroscope carefully to make sure to be coincident the rotation of z axis of gyro and the negative direction of the z axis of the body frame. Furthermore, as we assumed the vehicle has a rigid body, we treat the data from gyroscope as the rotation of body.

There is another problem of this system configuration. The transform between the body frame at the center of the rear axle and the camera frame located at the front wind shield. We approached to this problem as estimating the rotation and translation between two frames in the state

vector. The overall system configuration is shown in Figure 3.3.

3.2.1 State representation

We formed our system with gyroscope for rotations and wheel odometer for translation motion. Instead of using low-cost IMU like in original MSCKF paper, we choose odometer to provide reliable information to vision sensor. The gyroscope is modeled with bias and gaussian noise, and the wheel odometer is modeled with inverted scale factor λ and gaussian noise, which is tested and fixed before the experiment.

We assumed the modified system model as (3.31)

$$\mathbf{x}_{\mathrm{WO}} = \begin{bmatrix} \bar{\mathbf{q}}_{\mathrm{B}}^{\mathrm{G}} \\ \mathbf{p}_{\mathrm{GB}}^{\mathrm{G}} \\ \mathbf{b}_{\mathrm{g}} \end{bmatrix}, \qquad \tilde{\mathbf{x}}_{\mathrm{WO}} = \begin{bmatrix} \delta\boldsymbol{\theta}_{\mathrm{B}}^{\mathrm{G}} \\ \tilde{\mathbf{p}}_{\mathrm{GB}}^{\mathrm{G}} \\ \tilde{\mathbf{b}}_{\mathrm{g}} \end{bmatrix} \tag{3.31}$$

where $\bar{\mathbf{q}}_{\mathrm{B}}^{\mathrm{G}}$, $\mathbf{p}_{\mathrm{GB}}^{\mathrm{G}}$ are the unit quaternion of rotation and the position vector from global frame {G} and to body frame {B}, respectively. The velocity vector fell of the state since the information of velocity comes from the wheel odometer. The last term $\mathbf{b}_{\mathrm{g}}$ is the bias of gyroscope. As a result, the new state is constructed with 9 dimensions.

As we design to estimate the transformation matrix between the body frame and the camera frame, the totalized state includes calibration state.

$$\mathbf{x}_{calib} = \begin{bmatrix} \bar{\mathbf{q}}_{\mathrm{B}}^{\mathrm{C}} \\ \mathbf{p}_{\mathrm{CB}}^{\mathrm{C}} \end{bmatrix}, \qquad \hat{\mathbf{x}}_{calib} = \begin{bmatrix} \delta\boldsymbol{\theta}_{\mathrm{B}}^{C} \\ \tilde{\mathbf{p}}_{\mathrm{CB}}^{\mathrm{C}} \end{bmatrix} \tag{3.32}$$

Same with the MSCKF, we augmented the pose of the last N camera frames called *sliding window* state.

$$\mathbf{x}_{\mathrm{slw}_k} = \begin{bmatrix} \boldsymbol{q}_{C_k}^{G} \\ \boldsymbol{p}_{C_k}^{G} \end{bmatrix}, \quad \hat{\mathbf{x}}_{\mathrm{slw}_k} = \begin{bmatrix} \tilde{\boldsymbol{q}}_{C_k}^{G} \\ \tilde{\boldsymbol{p}}_{C_k}^{G} \end{bmatrix} \tag{3.33}$$

Finally, the overall state consist of the wheel odometer state, calibration state, and sliding window states.

$$\mathbf{x}_{\mathrm{slw}_k} = \begin{bmatrix} \boldsymbol{q}_{C_k}^{G} \\ \boldsymbol{p}_{C_k}^{G} \end{bmatrix}, \quad \hat{\mathbf{x}}_{\mathrm{slw}_k} = \begin{bmatrix} \tilde{\boldsymbol{q}}_{C_k}^{G} \\ \tilde{\boldsymbol{p}}_{C_k}^{G} \end{bmatrix} \tag{3.34}$$

3.2.2 System model and update

The system in continuous time is described in (3.35).

$$\begin{bmatrix} \dot{\bar{\mathbf{q}}}_{\mathrm{B}}^{\mathrm{G}} \\ \dot{\mathbf{p}}_{\mathrm{GB}}^{\mathrm{G}} \\ \dot{\mathbf{b}}_{\mathrm{g}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Omega\left(\mathbf{w}_{\mathrm{m}} - \mathbf{b}_{\mathrm{g}} - \mathbf{n}_{\mathrm{g}}\right)\bar{\mathbf{q}}_{\mathrm{B}}^{\mathrm{G}} \\ \mathrm{R}_{\mathrm{B}}^{\mathrm{G}}\lambda(\mathbf{v}_m - \mathbf{n}_{\mathrm{v}}) \\ \mathbf{n}_{\mathrm{bg}} \end{bmatrix} \tag{3.35}$$

where the vectors $\boldsymbol{w_m}(\mathrm{t})$ and $\boldsymbol{a_m}(\mathrm{t})$ are the data sequence along the time, $\mathbf{n}_{\mathrm{bg}}$ and $\mathbf{n}_{\mathbf{v}}$ are Gaussian random process. $\Omega(\boldsymbol{w})$ is defined as

$$\Omega(\boldsymbol{w}) = \begin{bmatrix} -[\boldsymbol{w}\times] & \boldsymbol{w} \\ -\boldsymbol{w}^T & 0 \end{bmatrix} \tag{3.36}$$

The data from gyroscope is modeled with bias and Gaussian noise, and the data from the wheel odometry is modeled with the inverted scale factor and Gaussian noise as (3.37).

$$\begin{aligned} \mathbf{w} &= \mathbf{w}_{\mathrm{m}} - \mathbf{b}_{\mathrm{g}} - \mathbf{n}_{\mathrm{g}} \\ \mathbf{v} &= \lambda(\mathbf{v}_{\mathrm{m}} - \mathbf{n}_{\mathrm{v}}) \end{aligned} \tag{3.37}$$

Since the body frame {B} is located at the center of the rear axle, the velocity vector $\mathbf{v}$ has only x component. Note that this is intentionally designed to apply non-holonomic constraint. (NHC)

Furthermore, the continuous model of the calibration state is (3.38)

$$\begin{bmatrix} \dot{\mathbf{q}}_{\mathrm{B}}^{\mathrm{C}} \\ \dot{\mathbf{p}}_{\mathrm{CB}}^{\mathrm{C}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \end{bmatrix} \tag{3.38}$$

From the definition of error state $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$, the model of our system is totalized as (3.39) and (3.40).

$$\begin{bmatrix} \dot{\tilde{\boldsymbol{\theta}}}_{\mathrm{B}}^{\mathrm{G}} \\ \dot{\tilde{\mathbf{p}}}_{\mathrm{GB}}^{\mathrm{G}} \\ \dot{\tilde{\mathbf{b}}}_{\mathrm{g}} \end{bmatrix} = \begin{bmatrix} -\hat{\mathrm{R}}_{\mathrm{B}}^{\mathrm{G}}\tilde{\mathbf{b}}_{\mathrm{g}} - \hat{\mathrm{R}}_{\mathrm{B}}^{\mathrm{G}}\mathbf{n}_{\mathrm{g}} \\ -\left[\hat{\mathrm{R}}_{\mathrm{B}}^{\mathrm{G}}\hat{\lambda}\mathbf{v}_{\mathrm{m}}\times\right]\tilde{\boldsymbol{\theta}}_{\mathrm{B}}^{\mathrm{G}} - \hat{\mathrm{R}}_{\mathrm{B}}^{\mathrm{G}}\hat{\lambda}\mathbf{n}_{\mathrm{v}} \\ \mathbf{n}_{\mathrm{bg}} \end{bmatrix} \tag{3.39}$$

$$\begin{bmatrix} \dot{\tilde{\boldsymbol{\theta}}}_{\mathrm{B}}^{\mathrm{C}} \\ \dot{\tilde{\mathbf{p}}}_{\mathrm{CB}}^{\mathrm{C}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \end{bmatrix} \tag{3.40}$$

Therefore,

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}_{\mathrm{WO}} &= \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & -\hat{\mathrm{R}}_{\mathrm{B}}^{\mathrm{G}} \\ -\left[\hat{\mathrm{R}}_{\mathrm{B}}^{\mathrm{G}}\hat{\lambda}\mathbf{v}_{\mathrm{m}}\times\right] & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}\tilde{\mathbf{x}}_{\mathrm{WO}} \\ &\quad + \begin{bmatrix} -\hat{\mathrm{R}}_{\mathrm{B}}^{\mathrm{G}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \hat{\mathrm{R}}_{\mathrm{B}}^{\mathrm{G}}\hat{\lambda} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \mathrm{I}_3 \end{bmatrix}\mathbf{n}_{\mathrm{IMU}} \\ &= \mathrm{F}\tilde{\mathbf{x}}_{\mathrm{IMU}} + \mathrm{G}\mathbf{n}_{\mathrm{IMU}} \end{aligned} \tag{3.41}$$

where $\mathbf{n}_{\mathrm{IMU}} = \begin{bmatrix} \mathbf{n}_{\mathrm{g}}^{\mathrm{T}} & \mathbf{n}_{\mathrm{v}}^{\mathrm{T}} & \mathbf{n}_{\mathrm{bg}}^{\mathrm{T}} \end{bmatrix}^{T}$.

The discrete state update, covariance update, state augmentation is similar with the process of the MSCKF except the Jacobian matrix $\boldsymbol{\mathcal{J}}_{\mathrm{WO}}$.

$$\mathrm{P}^{-} \leftarrow \begin{bmatrix} \mathrm{P}^{-} & \mathrm{P}^{-}\boldsymbol{\mathcal{J}}_{\mathrm{WO}}^{\mathrm{T}} \\ \boldsymbol{\mathcal{J}}_{\mathrm{WO}}\mathrm{P}^{-} & \boldsymbol{\mathcal{J}}_{\mathrm{WO}}\mathrm{P}^{-}\boldsymbol{\mathcal{J}}_{\boldsymbol{WO}}^{\mathrm{T}} \end{bmatrix} \tag{3.42}$$

$$\boldsymbol{\mathcal{J}}_{\mathrm{WO}} = [0_{6\times9} \quad \mathrm{I}_6 \quad 0_{6\times6N}] \tag{3.43}$$

This process is repeated every new camera frames are measured.

3.2.3 Measurement model and update

The measurement is also similar with the original MSCKF, except the measurement matrix and the measurement residual.

$$\mathbf{r}_i^{(\mathrm{j})} \simeq \mathrm{H}_{\mathbf{x}}^{(\mathrm{j})}\tilde{\mathbf{x}}^{-} + \mathrm{H}_{f_j}^{(\mathrm{j})}\tilde{\mathbf{p}}_{f_j} + \mathbf{n}_i^{(\mathrm{j})} \tag{3.44}$$

where the measurement matrixes $\mathrm{H}_{\mathbf{x}}^{(\mathrm{j})}$ and $\mathrm{H}_{f_j}^{(\mathrm{j})}$ are defined with the Jacobians as (3.45).

$$\begin{aligned} \mathrm{H}_{\mathbf{x}}^{(\mathrm{j})} &= \begin{bmatrix} \mathbf{0}_{2\times 9} & \mathbf{0}_{2\times 6} & \cdots & \boldsymbol{\mathcal{J}}_{\boldsymbol{i}}^{(\mathrm{j})}\left[\hat{\mathbf{x}}_{\boldsymbol{f_j}}^{\boldsymbol{C_i}}\times\right] & -\boldsymbol{\mathcal{J}}_{\boldsymbol{i}}^{(\mathrm{j})}\mathrm{R}_{\boldsymbol{G}}^{\boldsymbol{C_i}} & \cdots \end{bmatrix} \\ \mathrm{H}_{f_j}^{(\mathrm{j})} &= \boldsymbol{\mathcal{J}}_{\boldsymbol{i}}^{(\mathrm{j})}\mathrm{R}_{\boldsymbol{G}}^{\boldsymbol{C_i}} \end{aligned} \tag{3.45}$$

$$\boldsymbol{\mathcal{J}}_{\boldsymbol{i}}^{(\mathrm{j})} = \frac{\partial \mathbf{z}_i^{(\mathrm{j})}}{\partial \hat{\mathbf{p}}_{f_j}^{C_i}} = \frac{1}{Z_j^{C_i}}\begin{bmatrix} 1 & 0 & -\dfrac{X_j^{C_i}}{Z_j^{C_i}} \\ 0 & 1 & -\dfrac{Y_j^{C_i}}{Z_j^{C_i}} \end{bmatrix}$$

Chapter 4. New Method over RANSAC

4.1 Problem formulation

When pseudo-outliers dominate in an image, vision systems are hard to operate appropriately without some additional information. A framework using deep learning can be one of the solutions to this problem without additional sensors. However, it needs a huge amount of memory and computations, which is not our concern.

In this section, we describe the proposed method of choosing reliable indirect visual information with the additional sensors: inertial measurement unit (IMU) or wheel odometry. After we introduce a constraint in two-view geometry and the sources of errors, we describe a way to using the additional sensor regarding the sensor error analytically. The key idea is that the fused sensor can be a guideline of ego-motion. In order to select the best supportive group of features in the image, we apply k-nearest neighbor density estimation and mean shift algorithm to cluster and seek a proper mode. With the best supportive group, we use

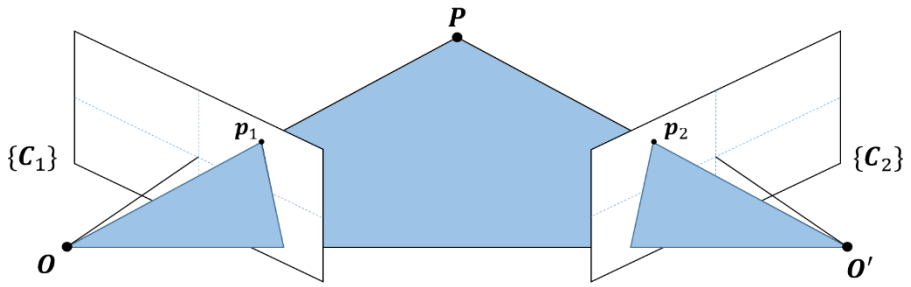


Figure 4.1. An illustration of Epipolar geometry constraint

the processed data with RANSAC to increase robustness. Finally, we summarize the overall method at **Algorithm 1**

At first, we focus on the Epipolar geometry constraint in two-view as described in Figure 4.1. $\boldsymbol{P}$ is a feature point, $\boldsymbol{O}$ and $\boldsymbol{O'}$ are the focal points at the first and the second viewpoint. $\boldsymbol{p}$ and $\boldsymbol{p'}$ are the projected points on each image plane. Epipolar geometry constraint means that feature point $\boldsymbol{P}$, focal points $\boldsymbol{O}$, and $\boldsymbol{O'}$ are placed on the same plane as Figure 4.1. The constraint is described as (4.1).

$$\boldsymbol{Op} \cdot (\boldsymbol{OO'} \times \boldsymbol{Op'}) = 0 \tag{4.1}$$

$$\boldsymbol{p}^T[\boldsymbol{t}\times]R_{C_2}^{C_1}\boldsymbol{p'} = d \approx 0 \tag{4.2}$$

In (4.2), $[\boldsymbol{t}\times]$is a skew-symmetric matrix of the transition vector $\boldsymbol{t}$ from C_1 to C_2 and $R_{C_2}^{C_1}$ is a rotation matrix from C_2 to C_1. Note that the matrix $E = [\boldsymbol{t}\times]R_{C_2}^{C_1}$ is called *fundamental matrix* in Epipolar geometry. Epipolar geometry constraint needs to be exactly zero,

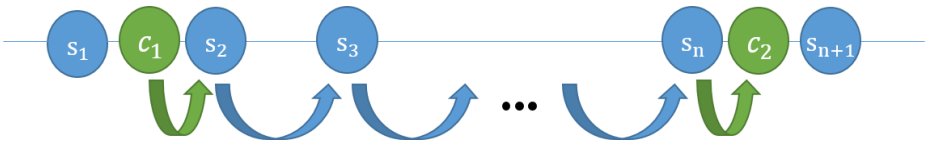


Figure 4.2. An integration of the state over time

however, the result of the real application is some non-zero value d. We called the value d as *Epipolar residual,* and the distribution of Epipolar residual d is the key element of our method. After replacing the error factors from the additional sensor, we can choose a group that has its Epipolar residual near zero. We considered three sources of error: an error of $\boldsymbol{t}$ from the wheel odometry, an error of $R_{C_2}^{C_1}$ from the gyroscope, and a tracking error between $\boldsymbol{p}$ and $\boldsymbol{p'}$ from two sequential images.

4.2 Local propagation

This process gets prior information from the additional sensors to calculate the Epipolar residual. We considered a system composed of IMU and wheel odometry using visual information as a measurement represented in the discrete state space. Using state propagation as described in section 3, we calculated the transformation from the current camera frame C_1 to the consecutive camera frame C_2. Due to the asynchronous sampling time of the additional sensor, it must be locally propagated between the camera states.

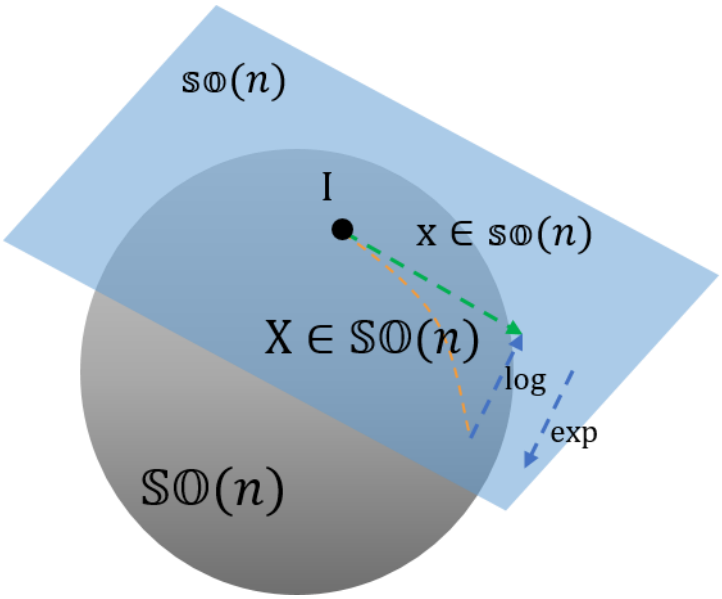


Figure 4.3. The relation between Lie group SO(n) and its tangent space

The word *local* is used to ensure that the propagation must be independent with the previous state with an initial condition.

$$\boldsymbol{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \qquad R_{C_1}^{C_1} = \mathrm{I}_{3\times 3} \tag{4.3}$$

We handled the data between $c_1$and s_2, s_n and c_2 carefully as shown in Figure 4.2. In order to avoid to integrate the propagation twice, preintegration [27] can be useful to make this process efficiently. The results are the rotation matrix $R_{C_2}^{C_1}$ and the translation vector $\boldsymbol{t}$.

4.3 Epipolar residual

Since we used a low-cost MEMS IMU, we assumed that the error from the gyroscope dominates the whole error and mainly considered the gyroscope error. We analyzed the error of the rotation matrix presented on manifold such as Lie group $SO(3) = \{X|XX^T = 1,\ det\ X = 1\}$ [29, 30]. The tangent space at the identity of $SO(3)$ is a Lie algebra $so(3) = \{x|x^T = -x\}$ as described in Figure 4.3. $so(3)$ is a vector space in $R\hat{}3$ with Lie bracket: a bilinear, antisymmetric bracket satisfying the Jacobi identity. In other words, any element in $so(3)$ can be represented by an orthogonal 3-dimensional vector as in (4.4)

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} \in \mathbb{R}^3, \qquad [\boldsymbol{\theta}\times] = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \in so(3) \qquad (4.4)$$

Since Lie group $X \in SO(3)$ and Lie algebra $x \in so(3)$ is related to the exponential mapping as $\exp(x) = X$, the output of the gyroscope modeled as $\boldsymbol{w} = \boldsymbol{w}_{ib}^{i} + \boldsymbol{w}_{bias} + \boldsymbol{w}_{noise}$ is represented as (4.5)

$$\exp([\boldsymbol{\theta}\times]) = \exp\left(\left[\boldsymbol{\theta}_{ib}^{i}\times\right] + [\boldsymbol{\theta}_{bais}\times] + [\boldsymbol{\theta}_{nosie}\times]\right) \qquad (4.5)$$

where $\boldsymbol{\theta} = \boldsymbol{w}\cdot\Delta t$ and $\boldsymbol{\theta}_{bias}$, $\boldsymbol{\theta}_{noise}$ and $\boldsymbol{\theta}_{ib}^{i}$ are rotation vectors of gyro bias, Gaussian noise and output of gyroscope from inertial frame to body frame represented in inertial frame respectively. Finally, we can represent the gyroscope element as in $SO(3)$. With *Rodrigues formula* [31] described in (4.6). We finally got the first order approximation of $\log|d|$ as in (4.7) including the pixel error $\Delta\boldsymbol{p}'$ from the moving object.

$$\exp([\boldsymbol{\theta}\times])$$

$$= \mathrm{I}_{3\times3} + \frac{\sin\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|}[\boldsymbol{\theta}\times] + \frac{(1-cos\|\boldsymbol{\theta}\|)}{2\|\boldsymbol{\theta}\|}[\boldsymbol{\theta}\times]^2 \tag{4.6}$$

$$\approx \mathrm{I}_{3\times3} + [\boldsymbol{\theta}\times] + \frac{1}{2}[\boldsymbol{\theta}\times]^2 \quad (when \quad \|\boldsymbol{\theta}\| \ll 1) \tag{4.7}$$

The first term of above equation represents the ideal Epipolar geometry constraint, which must be negative infinity. The rest of terms represent the pixel error due to some moving objects, the gyroscope bias error and the gyroscope noise error, respectively. If we could remove the effect of the gyroscope bias term appropriately, the existence of the second term highly effects the level of error. Note that the leftmost group in the $\log|d|$ space is the best supportive group and it is not on the negative infinity due to the noise and higher order terms.

$$\log|d| = \log\left|p^T\left([\boldsymbol{t}\times] + \left[\boldsymbol{t}_{obj}\times\right]\right)\exp([\boldsymbol{\theta}\times])\,\boldsymbol{p}'\right|$$

$$\approx \log\left|\boldsymbol{p}^T[\boldsymbol{t}\times]R_{C_2}^{C_1}\boldsymbol{p}'\right| + \log\left|\boldsymbol{p}^T[\boldsymbol{t}\times]R_{C_2}^{C_1}\Delta\boldsymbol{p}'_{obj}\right| \tag{4.8}$$

$$+ \log|\boldsymbol{p}^T[\boldsymbol{t}\times][\boldsymbol{\theta}_{bias}\times]\boldsymbol{p}'| + \log|\boldsymbol{p}^T[\boldsymbol{t}\times][\boldsymbol{\theta}_{noise}\times]\boldsymbol{p}'|$$

Additionally, the effect of the moving objects on the second term can be interpreted as an ego-motion with the opposite direction of the moving object described in (4.9).

$$\boldsymbol{p}^T\left([\boldsymbol{t}\times] + \left[\boldsymbol{t}_{obj}\times\right]\right)R_{C_2}^{C_1}\left(\boldsymbol{p}' + \Delta\boldsymbol{p}'_{obj}\right) = 0$$

$$\boldsymbol{p}^T\left([\boldsymbol{t}\times]R_{C_2}^{C_1}\right)\Delta\boldsymbol{p}'_{obj} \approx \boldsymbol{p}^T\left(\left[-\boldsymbol{t}_{obj}\times\right]R_{C_2}^{C_1}\right)\boldsymbol{p}' \tag{4.9}$$

The graphs illustrated in Figure 4.4. display histograms of an example

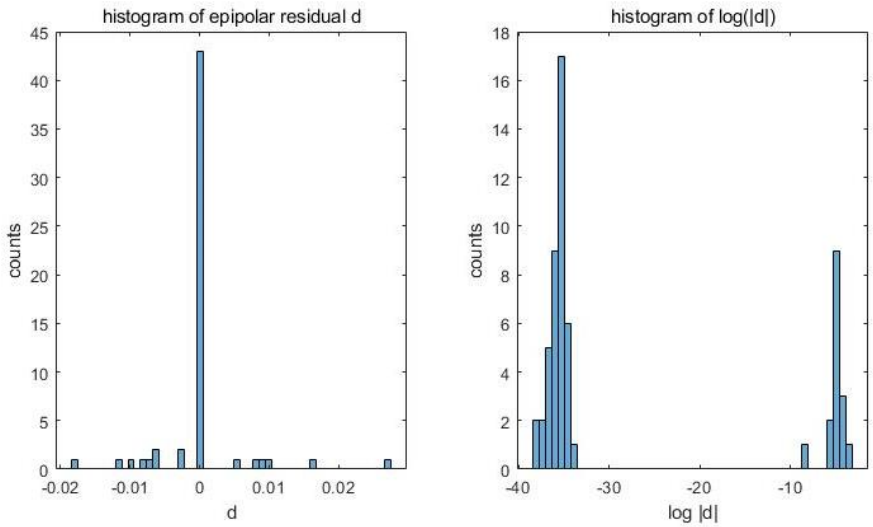


Figure 4.4. An example distribution of the epipolar residual d and log|d|

Epipolar residual data as a form of d and $\log|d|$ when there is a moving object on the way of the camera. In both of two graphs, we can distinguish the distribution of the Epipolar residual into two groups. However, the graph on the right is more clear to distinguish into a stationary feature group on the left and a moving feature group on the right. Therefore, we adapted the logarithm space for clustering.

4.4 Mode seeking

In order to distinguish the feature groups with modes, *Mean shift* [23, 28] was used as a representative algorithm for the mode seeking and clustering problem. Instead of the conventional mean shift with a distance threshold, we introduced modified weights from k-nearest

neighbor (k-NN) gaussian approximation as shown in (4.10). The mean shift calculates the weighted average of k-nearest points of each data point and shifts to the mean iteratively as in (4.11).

$$Let, S_j = \{k \ \ nearest \ \ neighbors \ \ of \ \ j\}$$

$$w_i = \sum_{j=1}^{N} L_j(i) \frac{X(\log|d_i|)}{\sum_{p=S_j} X(\log|d_i|)}, \qquad X \sim N\left(\sum_{p=S_j} \log|d_p|\right) \tag{4.10}$$

$$\log\left|d_i^{(n+1)}\right| = \frac{\sum_{p=S_j} \exp(w_i) \cdot \log\left|d_i^{(n)}\right|}{\sum_{p=S_j} \exp(w_i)} \tag{4.11}$$

where $L_j(i)$ is 1 when $i \in S_j$, otherwise 0. With large k, mode seeking is more robust at the cost of computation time.

In order to find the modes more efficiently, we used a non-maximum suppression (NMS) algorithm to merge the scattered modes. Since the naive mode shift algorithm needs a large number of iterations to gather the mode clearly, merging process applied on the modes helps to decrease computation time.

4.5 Clustering and sampling

Finally, we select the leftmost group and use the data group to calculate ego-motion. Figure 4.5. shows the result of mode seeking and clustering for an example data. For more robustness, we applied the 8-point RANSAC to the result. The overall algorithm is totalized at **Algorithm 1**.

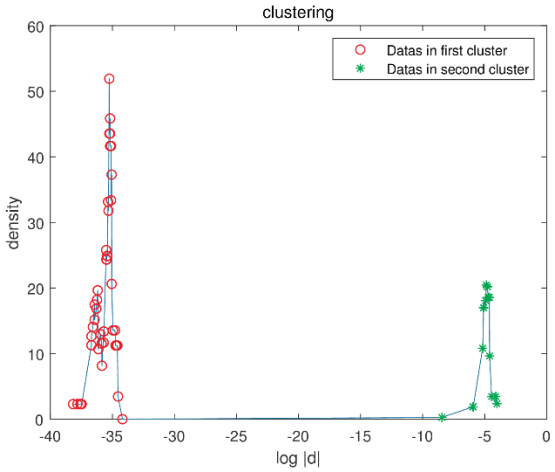


Figure 4.5. An example of mode seeking and clustering via mean shift

Algorithm 1 The details of the proposed algorithm
Result : A set of data points Local propagation with data from other sensors; Calculate $\log\|d\|$ for each data point; Initialize for each data point; ***While*** **(**Not converged)**{** Do mean shift with k-NN for each point; ***If*** **(**there are same data points)**{** Merge **&** Record index; **}** **}** Select the leftmost group of the data;

Chapter 5. Results

5.1 Simulation results

In this section, we performed a simulation to verify whether the proposed algorithm can filter out some non-stationary features, especially when there is a dominating pseudo-outlier group. An example simulation environment is demonstrated in Figure 5.1. We want to compare the result of our method and RANSAC, and finally to show it is operated well even if there are many of the moving features than the stationary features.

There is a robot with a camera follows a given path. We selected a group of features and made it move orthogonally to the path. The moment when the robot locates near the moving feature group is our interest. In order to evaluate without dependency on a specific system, we assumed we know about the motion of the camera. We applied, however, Gaussian noise on the position of p' and mismatch between p and p'.

In the left image of Figure 5.2., RANSAC regarded moving features as inliers improperly because the pseudo-outliers dominate. As a result, the ego-motion estimation based on the RANSAC exhibited that the camera goes to the left. On the contrary, in the right image, out method distinguished moving features based on the motion information and

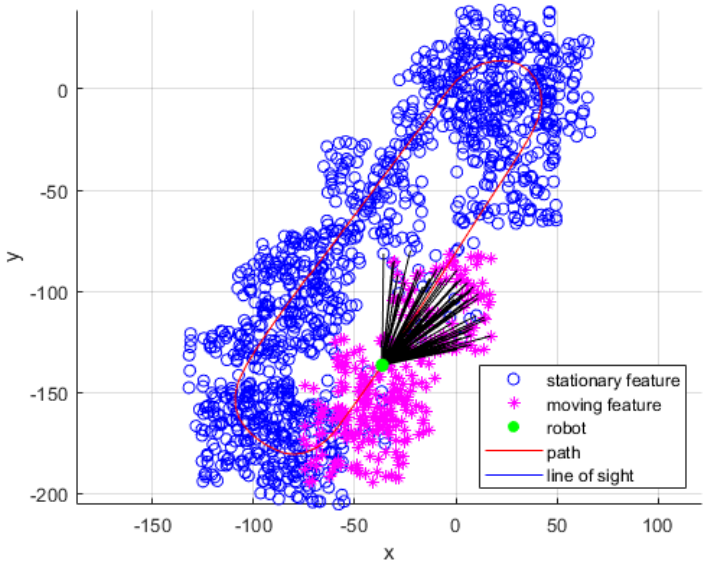


Figure 5.1. An example of the simulation environment

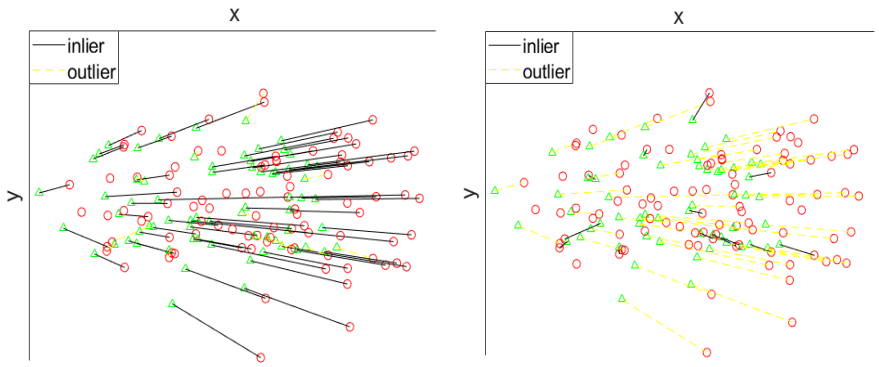


Figure 5.2. The outlier rejection results of RANSAC and our method

selected a few feature points as inlier. As a result, the ego-motion estimation based on our method exhibited that the camera goes straight. Except the case of all the detected points moves the same way, we can

see that the proposed method operated properly.

5.2 Experimental results

In this section, we described the result of the performance evaluation of the proposed method with a multi-state constraint Kalman filter with a hand-made dataset. The result of ego-motion estimation was improved than the conventional algorithm RANSAC, especially when there are some large moving objects in front of the camera.

5.2.1 System structure

Our system configuration is described in Figure 5.3. Although the MSCKF framework is originally designed for visual-inertial odometry, we used the framework with a gyroscope and wheel odometry since the accelerometer from the low-cost MEMS sensor is not appropriate to drive the system. We used the low-cost gyroscope with temperature compensation and heuristic drift reduction algorithm. The gyroscope and the wheel odometry are used for time update, and mono-camera is used for measurement update. Since the data from each sensor is not synchronized, we applied a time compensation algorithm. The position solution from GNSS is used as a reference. Table 5.1. shows the specifications of sensors.

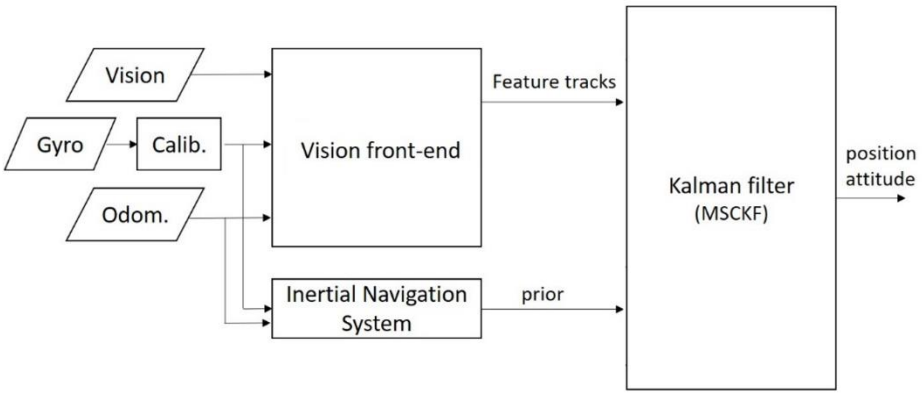


Figure 5.3. A summarized diagram of our system configuration

5.2.2 Dataset description

We collected data in various environments in Seoul, the central city of South Korea. This dataset contains unsynchronized image sequences of mono-camera, velocity data of each wheel, GNSS data with standard NMEA form, 3 DOF accelerometer, and 3 DOF gyroscope data from low-cost IMU. We manually designed embedded system used to collect the driving data. The system is installed on the front window of a test driving car as shown in Figure 5.4.

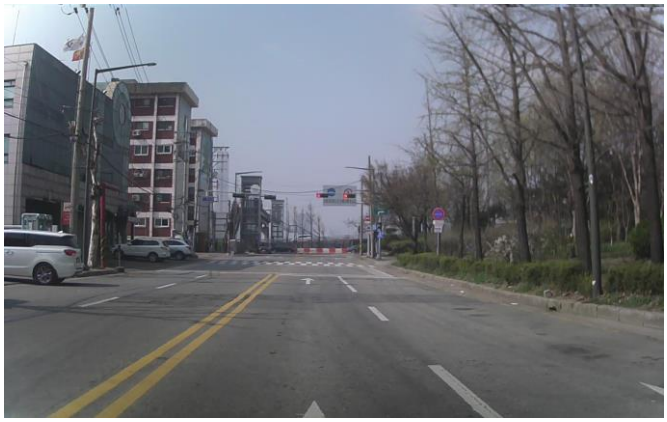
Unlike the KITTI dataset, there are many moving objects covering most of the view of the camera, including buses and trucks, as in Figure 5.5.

To make the evaluation test more analytically, we randomly choose some parts of the data sequence separated by the vehicle speed and the surrounding circumstance. For example, when the vehicle drives at high

Table 5.1. The specification of the sensors embedded in the system

Device	Content	Specification
Camera	Resolution	1280 X 800
	Frame Rate	10 fps
Gyroscope	Zero-point offset	$\pm\ 1deg/s$
	Offset variation	$\pm\ 1deg/s$
	RMS Noise	$0.02deg/s/\sqrt{Hz}$
	Sampling Rate	20Hz
Accelerometer	Zero-point offset	$\pm\ 70mg$
	Offset	$\pm\ 65mg$
	RMS Noise	$0.19mg/\sqrt{Hz}$
	Sampling Rate	20Hz

speed, tracking failure occurs more frequently. Furthermore, at the bridge or open area with only few objects, it is hard to track a reasonable amount of features. On the contrary, at tunnels or urban canyons, it is relatively easy to maintain a reasonable amount of tracking features.



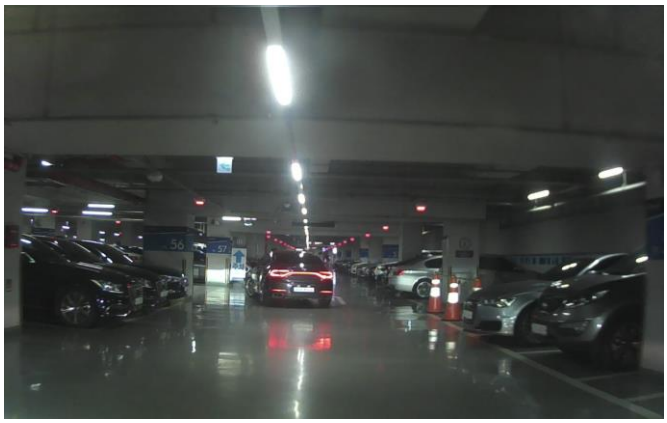


Figure 5.4. Example images of our dataset

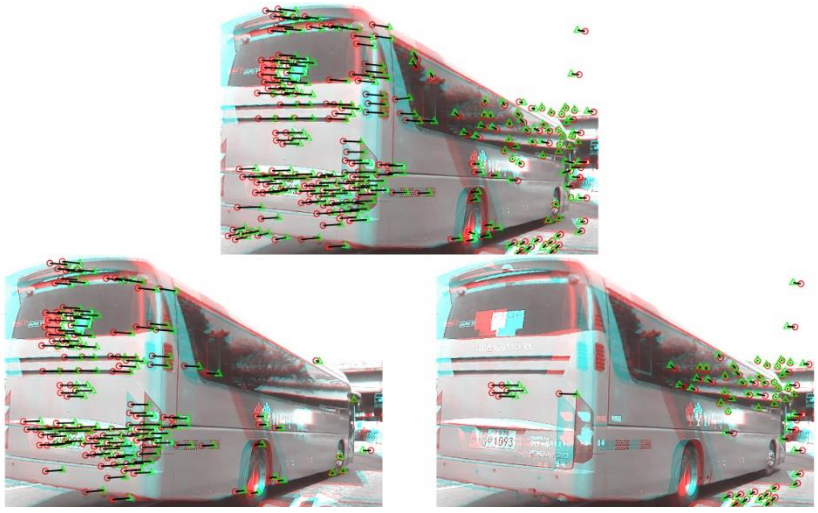


Figure 5.5. The result of RANSAC and our method with pseudo-oulier

5.2.3 Performance evaluation

We carefully searched for a suitable case making a significant difference between the RANSAC and our method. An example image in Figure 5.5. is the suitable case that we found in our dataset. A bus cut in front of our vehicle and occupied a large portion of the image. As shown in the bottom right of the image, our method operated more reliably than RANSAC.

We compared three cases: typical wheel odometry, MSCKF using 8-point RANSAC and MSCKF using our method. There is a rotational drift due to the low-cost MEMS gyroscope and incorrect ego-motion estimation, as illustrated in Figure 5.6. The event described in Figure 5.5.

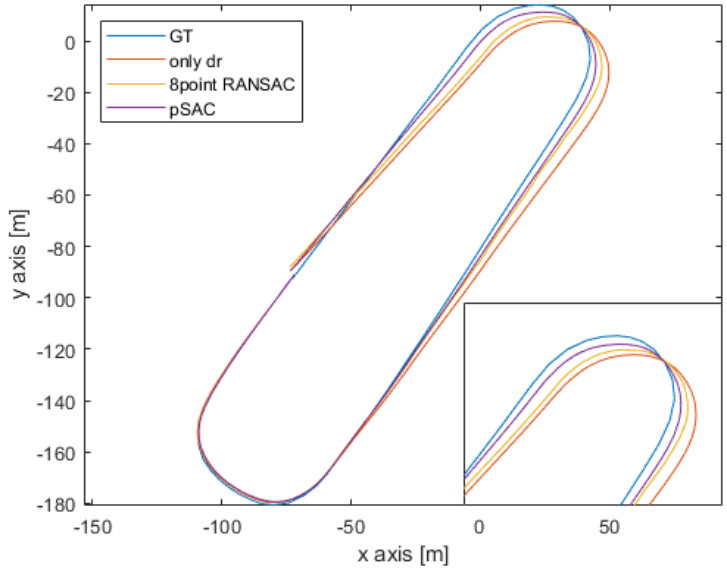


Figure 5.6. The estimated trajectory of ‘open area’ with various systems

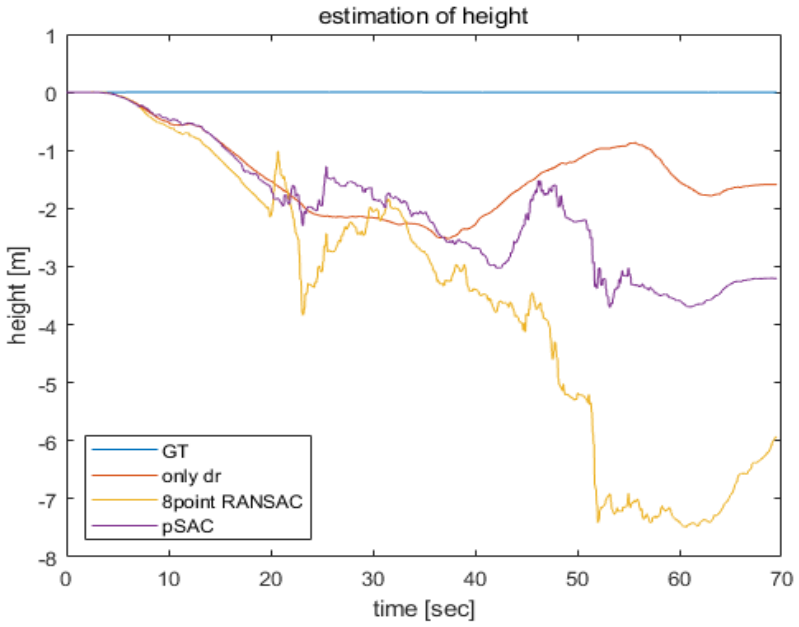


Figure 5.7. The estimated height of ‘open area’ with various systems

occurred at first corner, southeast of the graph. The trajectory of the 2-dimensional graph demonstrates that the visual information from our method is useful to improve the estimation of the target system. In terms of the estimation of the height, our method also displays a better result than RANSAC as shown in Figure 5.7.

We evaluate the performance of the odometry with mean squared error. The values on the Table is the percentage of error by the full length of each trajectory. Generally, the wheel odometry with vision performs better than the pure wheel odometry, as shown in the case 'urban canyon' in Table 5.2. However, since the case 'open area' has a situation causing malfunction of RANSAC, the performance of the MSCKF with RANSAC is even worse than the pure wheel odometry. On the contrary, wheel odometry with our method performs well as intended. In the case 'bridge', there were not many features on image and few rotations; therefore, the overall performance was decreased.

Table 5.2. The results of the odometrys evaluated with RMS

	Urban canyon	Open area	Bridge
Wheel	3.53%	0.96%	3.8%
Wheel + 8point	1.34%	1.00%	3.22%
Wheel + ours	0.75%	0.67%	3.22%

Chapter 6. Conclusion

6.1 Conclusion and summary

In this work, we proposed a method using an additional sensor to help to select the indirect visual information. Since there is a fundamental malfunction on the conventional method, the proposed method is designed to improve the disadvantage by evaluating and clustering. With the analysis of the error factors, we observed that the moving object in front of the camera makes a term of error. Since the gyroscope bias error and temperature nonlinearity is already compensated before the experiments, we expected that there is an explicit division along the ego-motion of feature. However, in contrast to the simulation, the Epipolar residual log|d| distributions of features overlap each other. In order to apply on the real application, we use density estimation and mode seeking method named kernel density estimation and mean shift respectively.

The method helps to get the piece of reliable information whenever a moving feature dominates on the image plane. As a results, it can avoid some situations that conventional algorithm only depends on the vision data confused totally, except the case that all the field of view is covered by a moving object so the estimation of ego-motion totally goes wrong. The performance of our method through simulation and real-world

experiments, and verified that the proposed method can improve the performance of vision-aided navigation system. Especially when there is a large moving object as the case of the bus, our propose method even increases the estimation accuracy.

6.2 Future works

There are some ways to develop this work more diverse. As we dominantly consider about a large moving object, there is a way to research about the multiple objects. Furthermore, we can think about the other space and selection method.

Although we used 8-point RANSAC to the result of our method to increase the robustness, this method suffers from the wreck of the number of features. If we can treat the features of each moving object, not the largest one, this can be applied to select only the ego-motion.

Many of clustering methods for multiple group data, applying the propose concept at the clustering process instead of the mean shift. Moreover, the more efficient space transformation instead of log|d| will be the most important part of the next researches. Coupling tightly with the MSCKF, another rejection algorithm can be studied with the stacked features, not limited to the two view Epipolar geometry. However in this case, the stacking error from the noise of sensor must be managed properly.

Bibliography

[1] M.A. Fischler, “Random Sample Consensus,” in *Proceedings of the 26th Conference on Decision and Control*, 1987, pp. 98-99.

[2] A.I. Mourikis and S.I. Roumeliotis, “A Multi-State Constraint Kalman Filter for Vision-Aided Inertial Navigation,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007.

[3] Mark Maimone, Yang Cheng and Larry Matthies, “Two Years of Visual Odometry on the Mars Exploration Rovers,” *Journal of Field Robotics*, vol. 24, pp.169-186, Mar. 2007.

[4] Yang Cheng, Mark Maimone and Larry Matthies, “Visual Odometry on the Mars Exploration Rovers,” in *2005 IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2005.

[5] T. Qin, P. Li and S. Shen, “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator,” in *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004-1020, Aug. 2018.

[6] S. Heo, J. Cha and C. G. Park, “EKF-Based Visual Inertial Navigation Using Sliding Window Nonlinear Optimization,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2470-2479, Jul. 2019.

[7] M. Schreiber, H. Königshof, A. Hellmund and C. Stiller, “Vehicle Localization with Tightly Coupled GNSS and Visual Odometry,”

2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, 2016, pp. 858-863.

[8] R. I. Hartley, "In Defense of the Eight-Point Algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, Jun. 1997.

[9] Davide Scaramuzza and Friedrich Fraundorfer, "Tutorial: Visual Odometry," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, Dec. 2011.

[10] P.H.S. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Computer Vision and Image Understanding*, vol. 78, pp.138–156, 2000.

[11] P.H.S. Torr, "Bayesian Model Estimation and Selection for Epipolar Geometry and Generic Manifold Fitting," *International Journal of Computer Vision*, vol. 50, no. 1, pp.35–61, 2002.

[12] Ben J. Tordoff and David W. Murray, "Guided-MLESAC: Faster Image Transform Estimation by Using Matching Priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp.1523–1535, Oct. 2005.

[13] D.R. Myatt, P.H.S Torr, S.J. Nasuto, J.M. Bishop, and R. Craddock, "NAPSAC: High Noise, High Dimensional Robust Estimation - It's in the Bag," in P*receedings of the 13th British Machine Vision Conference (BMVC)*, pp. 458–467, 2002.

[14] J. Matas and O. Chum, "Randomized RANSAC with Sequential Probability Ratio Test," in *Proceedings of the 10th IEEE*

International Conference on Computer Vision (ICCV), 2005.

[15] A. Konouchine, V. Gaganov and V. Veznevets, “AMLESAC: A New Maximum Likelihood Robust Estimator,” in *Proc. Graphicon*, 2005.

[16] T.K. Moon, “The Expectation-Maximization Algorithm,” *IEEE Signal Processing Magazine*, vol. 13, no. 6, Nov. 1996.

[17] S. L. Choi, T.M. Kim, and W.P. Yu, “Performance Evaluation of RANSAC Family," *The Technical Writer's Handbook*, Mill Valley, Seoul, 1989.

[18] C.V. Stewart, “Bias in Robust Estimation Caused by Discontinuities and Multiple Structures,” *Pattern Analysis and Machine Intelligence*, 2009.

[19] M. Zuliani, C.S. Kenney, B.S. Manjunath, “The multiRANSAC Algorithm and Its Application to Detect Planar Homographies,” *IEEE International Conference on Image Processing 2005*, Sep. 2005.

[20] Toldo R. and Fusiello A., “Robust Multiple Structures Estimation with J-Linkage,” *European Conference on Computer Vision (ECCV) 2008*, Oct. 2008.

[21] Luca Magri and Andrea Fusiello, “T-Linkage: A Continuous Relaxation of J-Linkage for Multi-Model Fitting,” *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3954-3961.

[22] Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[23] Yizong Cheng, “Mean Shift, Mode Seeking, and Clustering,” *Journal IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, Aug. 1995.

[24] H. Isack and Y. Boykov, “Energy-Based Geometric Multi-Model Fitting,” *International Journal of Computer Vision*, vol. 97, pp.123-147, Apr. 2012.

[25] A. Geiger, P.Lenz, C. Stiller, and R. Urtasun, “Vision Meets Robotics: The KITTI Dataset,” *International Journal of Robotics Research (IJRR)*, 2013.

[26] Michael Burri et al., “The EuRoC Micro Aerial Vehicle Datasets,” *International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157-1163, Sep. 2016.

[27] C. Forster, L. Carlone, F. Dellaert and D. Scaramuzza, “On-Manifold Preintegration for Real-Time Visual-Inertial Odometry,” *in IEEE Transactions on Robotics*, vol. 33, no. 1, pp.1-21, Feb. 2017.

[28] D. Comaniciu and P. Meer, “Mean shift: A Robust Approach toward Feature Space Analysis,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, May 2002.

[29] A. Kirillov, Jr, *An Introduction to Lie Groups and Lie Algebras,* Cambridge University Press, 2008.

[30] E. Eade, “Lie Groups for 2D and 3D Transformations,” [Online]. Available: http://ethaneade.com/lie.pdf. [Accessed Nov. 25, 2019].

[31] Jian S. Dai, “Euler–Rodrigues Formula Variations, Quaternion Conjugation and Intrinsic Connections,” *Mechanism and Machine*

Theory, vol. 92, pp.144-152, Oct. 2015.

[32] M. Himmelsbach, A. Müller, T. Lüttel and H.-J. Wünsche, "LiDAR-Based 3D Object Perception," *Proceedings of 1st International Workshop on Cognition for Technical Systems*. vol. 1, 2008.

[33] Sen Wang, Ronald Clark, Hongkai Wen, Niki Trigoni, "DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks," *arXiv:1709.08429v1* [cs.CV], Sep. 2017.

[34] (31 March 2018) Tesla in Fatal California Crash Was on Autopilot. BBC NEWS. *Retrieved from* https://www.bbc.com/news/world-us-canada-43604440.

[35] Christian Kerl, Jürgen Sturm, and Daniel Cremers, "Deep Visual SLAM for RGB-D Cameras," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013.

[36] Hartley, Richard, and Zisserman, Andrew, *Multiple View Geometry in Computer Vision*, Cambridge University Press, New York, NY, USA, 2003.

초 록

이 논문에서는 실제적인 영상 보조 항법 시스템에 적용하기 위한 알고리즘을 제안하였다. 영상 보조 시스템에서 가장 많이 사용하는 무작위 추출 컨센서스의 경우 움직이는 물체와 같은 수도-아웃라이어 때문에 실제 세상에 적용하기 힘들다. 제안한 알고리즘은 이와 같은 기존 방법의 문제를 해결하고자 고안되었으며, 시스템을 운용하는 센서로부터 사전 정보를 받아 이용한다. 큰 물체가 화면의 대부분을 가리는 경우에 적절하게 행동하는지 판별하기 위해 시뮬레이션 시행하여 확인하였다. 제안된 알고리즘을 검증하고자 휠 오도메트리와 자이로스코프 및 영상을 결합한 다중상태조건 칼만 필터 시스템을 이용하여 실제 취득한 데이터에 적용하였다. 기존 알고리즘이 오동작하는 수도-아웃라이어가 있는 경우에도 제안한 알고리즘은 적절하게 동작하였으며, 궤적 추정의 정확도가 향상되는 것을 확인하였다.

주요어: 영상보조시스템, 수도−아웃라이어, 다중상태조건 칼만 필터, 랜덤 샘플 컨센서스, 평균점 이동 클러스터링
학 번: 2018-22641