Lehrstuhl für Kommunikation und Navigation
Technische Universität München

Cooperative Vision for Swarm Navigation

Chen Zhu

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Eckehard Steinbach
Prüfer der Dissertation:	1. Prof. Dr. sc. nat. Christoph Günther
	2. Assoc. Prof. Michael Kaess, Ph.D.

Die Dissertation wurde am 11.06.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 31.10.2019 angenommen.

Abstract

Mars has become a focal point of space in-situ exploration. The associated exploration tasks are currently performed using large tele-controlled rovers. In the future these large rovers shall be complemented by swarms of small rovers and crawlers that are operating in a much more independent manner. Positioning is needed both for navigating the exploring vehicles and for referencing observations. The current lack of infrastructures makes these tasks challenging. The aim is thus to enable positioning with minimum means, e.g. using the lander and a sparse network of anchor stations.

Cameras play an essential role in this context. By using stereo cameras, mounted on the rovers, the egomotion and a map of the environment can be estimated simultaneously. This is called (visual) simultaneous localization and mapping (SLAM). Consequently, a rover can create a reference frame, i.e., navigation frame, with the origin at the starting position, and represent its trajectory and map in the frame by applying SLAM algorithms. The transformation from the navigation frame to a global reference frame can be resolved by using anchor points with known location in both frames. However, the accumulation of errors in the camera tracking strongly limits the accuracy of the positioning and the mapping, except if loop closures can be established. The latter constrains the trajectories and slows down the exploration. Additionally, the stereo camera rigs, used in the present context, require large baselines for accurately estimating the system scale and thus the true position. Such rigs cannot be adapted to small rovers. Monocular systems are more appropriate on such rovers.

We solved the challenges posed by the use of monocular cameras and the lack of loop closures in the exploration paths by a sensor fusion algorithm based on monocular pictures and ranges to a single anchor point, e.g., the landing site. Unlike conventional radio-based positioning, the present work shows the potential associated with the combination of a single wireless link connection and pictures from a monocular camera.

With a swarm of robots equipped with monocular cameras and communication links, ranges can be determined between each pair of rovers. Any static rover can thus act as an anchor for the aforementioned sensor fusion method. In the case of moving robots, we propose a more advanced scheme. It performs relative pose and scale estimation and achieves a further improved performance. Unlike conventional map merging approaches, our method does not require the exchange of images or feature descriptors. As a consequence, the data rates are low. In addition, the estimation of the relative pose allows for multi-view vision with a large baseline. In the case, of two rovers, the resulting virtual stereo rig can be larger by order of magnitude as compared to a physical stereo rig. We also developed a common field of view detection algorithm for cooperative scene reconstruction in order to take advantage of these capabilities.

We also developed a swarm navigation solution based on pose graph optimization by applying the proposed relative pose estimation. We considered both centralized and distributed solutions. Since the cooperative pose estimation requires data exchange among rovers, the communication requirements increase with the number of rovers in the swarm. As a consequence, we developed an adaptive swarm grouping method, which takes both geometric distance and common field of view of cameras as the metric. This allows to significantly reduce the communication load in the system.

Besides for Mars, the proposed methods can also be applied and extended to other scientific missions as well as to terrestrial applications, such as autonomous car-driving, disaster rescues, and cooperative mapping.

Acknowledgment

Curiosity is the inner fuel that drives us to keep thinking and to explore the world. Driven by curiosity, the universe always look so mysterious and attractive to me. If we look from the scale of the universe, the whole history of humankind is just a blink of an eye. From this perspective, the time for pursuing and getting the doctoral degree does not seem so long any more. I would like to thank many people for their help and supports over the whole period.

First of all, I would like to express my gratitude to my supervisor Prof. *Christoph Günther* for his consistent supports on my research in every sense of the word. I really appreciate the freedom he gives me on the research, which is significantly important for innovative work.

I also want to show my most sincere appreciation to Prof. *Michael Kaess* from CMU for his patience and helps throughout the whole process. I cannot express more gratitude for squeezing the work related to my thesis into his full schedule and for our talks during the very short visits.

Special thanks to my colleague and friend *Gabriele Giorgi*. Our precious discussions have inspired me to develop one the the core innovative points of the thesis.

I appreciate a lot the joint work on the visual navigation topics with *Christoph Bamann* and *Young-Hee Lee*. It is nice to have discussions and work together.

Important appreciations to other colleagues/friends of the NAV chair at TUM, including *Zhibo Wen*, *Andreas Brack*, *Sebastian Knogl*, *Patrick Henkel*, *Martin Lülf*, and *Kaspar Giger*. I really treasure the time we spent every day together during the years.

I am also grateful to the colleagues working on the VaMEx projects together from DLR, TUM, and TU-BS, especially to *Siwei Zhang* for our inspiring discussions on research over years.

Many thanks to the close colleagues at DLR for backing me up when I was finalizing my dissertation. Particularly, I am very grateful to Prof. *Michael Meurer* for his strong supports on further developing the related topic in the department. Thanks to *Thilo Schuldt* for giving me precious suggestions on revising the German abstract.

Acknowledgment to the great students I have been working together with and to other colleagues and good friends who are supportive of my pursuing of the degree.

Last but most important, I would like to thank my family in Chinese:

感谢我的父母和祖父母多年来对我的养育之恩！没有你们的教育，就不会有今天的我。感谢我的叔叔婶婶在德国留学这些年对我的莫大帮助与支持！感谢我最亲爱的妻子对我的理解，支持以及付出！我的博士学位跟你们的爱和付出密不可分。还有其他所有支持我的家人们，这里虽不一一历数，但你们对我的耐心与包容我都感恩在心。

朱宸

Contents

List of Symbols

$\lfloor\cdot\rfloor_\times$	skew symmetric matrix constructed from a 3D vector
$\vec{\cdot}$	vector with geometric meaning in Cartesian coordinates
$\tilde{\cdot}$	vector with geometric meaning in Homogeneous coordinates
$.^{(\cdot)}$	superscript denoting the reference frame of a point
$\mathbf{j}.$	superscript on the left denoting the rover index in a swarm
$\cdot_{[k]}$	subscript with square brackets denoting the time index of a variable
(W)	global reference frame
(N)	navigation reference frame
(N_j)	navigation reference frame of rover j in a swarm
(C)	camera body reference frame
$R_{(P\to Q)}$	rotation between reference frame (P) and frame (Q)
$\vec{t}_{(P\to Q)}$	translation between reference frame (P) and frame (Q)
$\vec{O}$	origin of a reference frame
$\vec{X}$	point location in 3D space
$\vec{c}$	camera location in 3D space
$\vec{\beta}$	location in 2D planar motion
ϕ	attitude (heading angle) in 2D planar motion
x	pose vector including both position and attitude parameters
f	camera focal length
d	depth of a point in the space
K_C	camera intrinsic matrix
P	camera projection matrix
$\vec{b}$	baseline between two cameras in a stereo rig
$\pi(\cdot)$	projection function which projects a 3D point to a 2D location in the camera image plane
$\pi^{-1}(\cdot)$	function which back-projects a 2D point in the camera image plane to 3D space
u_i	2D feature location of point i in the image plane
μ_i	noisy measurements of u_i
r	true range
ρ	noisy ranging measurement of r
Σ	covariance matrix
ξ	vector containing parameters to be estimated
J	Jacobian matrix
s_g	global scale factor
s_r	relative scale factor
θ	relative heading angle between two coplanar cameras
α	azimuth angle of a 2D point in polar coordinates
N_p	number of points
N_k	number of keyframes
N_r	number of rovers in a swarm
m	index of class in grouping and clustering
M	total number of classes in grouping and clustering
λ	eigenvalue of a matrix
$\tilde{\gamma}$	plane equation parameterized by homogeneous coordinates

1. Introduction

Mars exploration is an exciting and meaningful mission for scientists and the whole human race. The information and knowledge obtained by exploring Mars can guide us to a better understanding of the solar system, the earth, and possibly life. The exploration tasks include the measurement of physical quantities on Mars, surface terrain mapping, detection of water and biochemical hints, etc. The tasks require Mars rovers that are capable of collecting sensor measurements and scientific samples in various landscapes such as plains, canyons and caves, and the position of the rovers should be available with respect to a global reference frame. However, unlike the condition on the earth with global navigation satellite systems (GNSS), the possible infrastructures to position the Mars rovers are rather restricted. Only the orbiters and the landing site can serve as reference infrastructures with known absolute coordinates on Mars. Nevertheless, the visible time of the orbiters is limited due to the low number of satellites with long revisit period, and a rover will lose line-of-sight connections to the landing site when it travels far away from the landing point. In such lack-of-infrastructure condition, camera plays an important role not only in perception and mapping, but also in autonomous positioning and navigation of the robots. By applying visual cues, a rover is capable of positioning itself by matching with known landmarks in the existing map. Moreover, a moving robot can accomplish egomotion estimation using cameras according to the change in the images over time, namely the visual odometry (VO) technique. Maimone et al. has shown in [1] that in the past Mars rover mission, visual odometry has significantly outperformed conventional wheel odometer, since the rugged surface full of sands significantly affects the performance of a wheel odometer. Meanwhile, cameras can build a three-dimensional (3D) map during sensing, while its location in the map is estimated. This enables the exploration approach of the simultaneous localization and mapping (SLAM), which estimates the trajectory of the vehicle and simultaneously reconstructs the map of the sensed environment.

However, as a dead reckoning system, the error of visual odometry accumulates over time, so the estimated trajectory will drift from the true one. To diminish the accumulation of error, loop closure techniques are proposed in visual simultaneous localization and mapping (VSLAM) researches. The camera can detect and recognize some visual cues which already exists in the map the robot has built. As a result, the whole estimated trajectory and the map can be corrected by closing the measurement loop, since the errors are highly correlated temporally. However, the loop closure technique requires the vehicle to revisit some already mapped place, which significantly degrades the exploration efficiency. Both the error accumulation and the efficiency problem become more crucial if the mission plans to explore an immense region such as Valles Marineris, the 4,000-km-long canyon on Mars with enormous scientific attractions to be investigated [2]. In addition, the remote control mode of the rover is impractical for such large-scale exploration missions. Due to the long distance between Mars and Earth, it takes a few minutes for signal one-way propagation even at speed of light. Remotely controlling a rover from earth to investigate a large area may take ages, not to mention the radio connection cannot be consistently retained given the few infrastructure. As a potential solution to the challenges, researchers from German Aerospace Center (DLR) propose to use an autonomous and cooperative robotic swarm of independent rovers, crawlers and potentially flying platforms to explore Valles Marineris [3]. Fig. 1.1 shows the concept of Mars exploration using robotic swarms.

A cooperative robotic swarm with communication links among each other has the following advantages compared with a single rover:

1) Different regions can be explored in parallel using a swarm, so that the efficiency of information collection is significantly enhanced;

2) The observability of the perception sensors, e.g., cameras, is better in the swarm case, since occlusions have smaller impact thanks to the independent mobility of individual platforms;

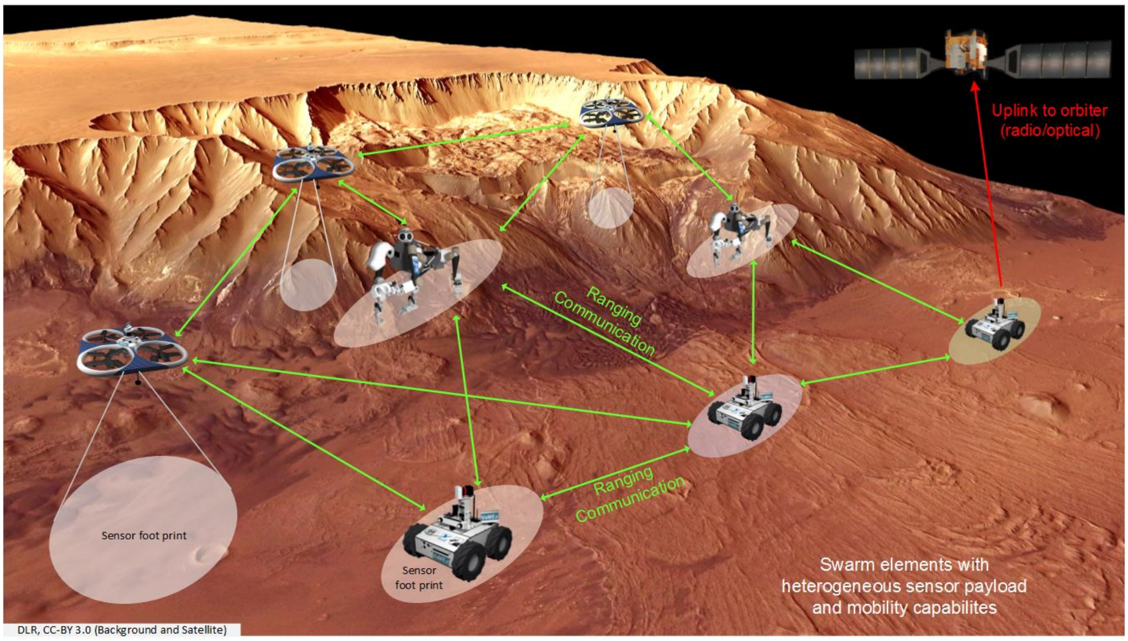


Figure 1.1: Mars Marineris exploration using robotic swarms [4]

3) In terrains such as caves and canyons, the radio connection will be interrupted for a single rover, while the communication and line of sight connection can still be retained in swarm case by using another rover as relay;
4) The swarm elements can form a sensor network and jointly position themselves using ranging measurements from wireless radio (for instance, applying the methods described in work from Staudinger et al. [5] and Zhu et al. [6]);
5) The formation of the swarm is variable, so that better geometry can be achieved by controlling the rovers both in swarm positioning tasks (as shown in work from Zhang et al. [7]) and in camera-based multi-view mapping tasks;
6) A robotic swarm is robust against failure of a single platform, because the whole mission normally can tolerate certain failures in one or a small portion of the swarm elements.

On the other hand, the design of individual platforms for a robotic swarm must be miniaturized, given the constraints on cost, size, and weight for the mission. This brings new challenges to the swarm navigation. Compared with single-rover-based approaches, a swarm element in ordinary has much smaller size, so that the payload, the storage and the computational power is more limited. Hence in most cases either it cannot afford to equip a stereo camera rig, or a stereo rig has barely advantage comparing with a monocular camera due to the very short baseline and the limited resolution. As a result, the design of the visual navigation methods for a swarm should be based on the monocular camera assumption. Since the depth information of the captured scenes is lost in perspective projection, the three-dimensional (3D) position of the points of interest cannot be reconstructed without ambiguity using a single camera. The estimated trajectories and maps using a monocular camera have a global scale ambiguity. Although the relative scale between two ego-motion steps can be estimated using a camera, the product of these relative scales accumulates the estimation error exponentially. This is the well-known challenging scale drift problem in monocular VSLAM. The topic of scales will be explained in more details in Section 4.1. In addition, the rovers in a swarm need to jointly navigate themselves and observe scenes of interest, and to coordinate the exploration task cooperatively (such as in work from Wiedemann et al. [8]). As a result, the swarm navigation requires not only the navigation of a single rover, but also the relative positioning between swarm elements.

To provide a solution for the above challenges, this thesis proposes to apply sensor fusion for swarm navigation using monocular cameras and ranging measurements between swarm elements. The range estimates can be obtained from round trip delay (RTD) estimation exploiting pilot signals in the mobile communication channel, which is available between any pair of vehicles due to the cooperative essence among swarm elements. In addition, by exchanging basic information, the swarm can outperform a single platform in accuracy by jointly positioning the rovers. The main contributions of the thesis are:

1) A global scale estimation method using an onboard monocular camera and ranging measurements from the communication link to a single static station, e.g., another rover in static mode;
2) A tight coupling sensor fusion algorithm using a monocular camera and sparse ranging measurements from a static station. By fusing with the ranging measurements, the error of the visual SLAM is bounded at a low level without requiring to revisit mapped locations for loop closure;
3) An innovative relative pose and scale estimation method for two dynamic rovers equipped with monocular cameras and connected by a communication link;
4) It is shown that the visual SLAM accuracy can be significantly improved by tightly coupled fusion of the visual and ranging measurements from the two cooperative rovers;
5) A swarm navigation solution based on pose graph optimization by applying the proposed relative pose estimation. Both centralized and distributed solutions are considered;
6) A common field of view detection algorithm for two independent cameras;
7) An adaptive swarm grouping method, which takes both geometric distance and common field of view as metric. By applying the grouping strategy, the intra-group communication load can be bounded to a feasible level.

Such methods can be equally applied or easily adapted on a number of applications in terrestrial environments. For example, Scaramuzza et al. propose in [9] to use a swarm of vision-controlled microaerial vehicles (MAVs) to execute searching, surveillance, and rescue missions in GNSS-denied environments. In addition, the infrastructures on the earth provides more possibilities to obtain ranging measurements. For example, the cellular networks provide pilot signals for ranging using time of arrival (TOA) estimates. The widely used long-term evolution advanced (LTE-A) system proposes and analyzes device-to-device communication channels in [10] since specification release 12. In the standard [11], Fine Timing Measurement is standardized for WiFi radio networks. Moreover, in the upcoming fifth-generation of wireless mobile telecommunications technology (5G), device-to-device communication, which has ranging capability, is thoroughly analyzed, for instance by [12] and plays a significant role in mobile communication systems in the future. Therefore, the methods of swarm visual navigation can also potentially provide a solution for terrestrial applications such as the positioning of connected vehicle networks on the road with car-to-car communications and ranging links, especially in the GNSS-degraded urban canyons. Despite the variety of the application scenarios, in order to keep representative and consistent, the following chapters of this work will be introduced based on the typical Mars exploration scenario.

This dissertation is organized as following: In Chapter 2, the sensor measurement models and the system model are introduced. Then, the visual navigation problem using visual sensor only, e.g., a stereo camera rig, on an individual vehicle is discussed in Chapter 3. The visual SLAM framework is reviewed and a general spatial uncertainty model is introduced. According to the uncertainty analysis, the reason why monocular cameras are preferred in our application is explained. In Chapter 4, the scale estimation problem in monocular visual navigation is explained, and we propose a method that jointly estimates the global scale of the monocular camera and the relative pose between a dynamic rover and a static station by exploiting visual measurements from monocular cameras and sparse ranging measurements from a radio link. Based on that, a tight coupling sensor fusion method is proposed which outperforms the vision only approach in SLAM but with similar computational complexity. Chapter 5 discusses the scenarios of two dynamic rovers. It is shown that the relative pose between two dynamic rovers can be estimated without any scale ambiguity by using the monocular cameras and the ranging measurements from the radio link between the two rovers. Furthermore, the egomotion estimation accuracy can be improved by exploiting the data from the two rovers using our tight coupling sensor fusion method. The approaches do not demand to transmit any image data or feature descriptors, which has relatively weak demands on the communication bandwidth. In addition, a common field of view detection algorithm is proposed to aid the 3D reconstruction using two monocular cameras mounted on different rovers. As an extension from a pair of vehicles, swarm navigation methods are proposed in Chapter 6. The swarm pose estimation methods using sensor fusion for both centralized mode and distributed mode are discussed. Additionally, as a solution to the scalability problem of large robotic swarms, an adaptive grouping algorithm is proposed by considering both relative geometry information and common field of view of the cameras on different rovers. At the end of the dissertation, a conclusion of this work is drawn. The simulation and test results are shown directly in each corresponding section.

2. Measurement Models and System Models

The swarm system being discussed consists of several autonomous vehicles equipped with cameras and radio front-ends with ranging capability. The models of both optical measurements and range measurements are described in this chapter. Moreover, the detailed model of the autonomous swarm is provided.

In the remainder of this dissertation, a superscript with parentheses $^{(\cdot)}$ is used to denote the reference frame in which the vector is represented. Vectors such as $\vec{c} \in \mathbb{R}^3$ with geometric meanings are written with an arrow. Time, denoted with square brackets $[\cdot]$, is measured in keyframes, i.e., the time reference instances in which both the range measurements and the trajectory estimation are available. The homogeneous coordinates in the extended Euclidean plane are written as $\tilde{u} \in \mathbb{P}^2$. For scenarios with multiple robots in a swarm, a superscript j before a variable denotes the index of the robot. Moreover, this work uses $[A, B]$ and $[A|B]$ to denote the horizontal concatenation of two matrices, and uses $[A; B]$ to denote the vertical concatenation.

2.1 Reference Frame Transformation

Generally, the transformation between two three dimensional (3D) reference frames (P) and (Q) follows

$$\vec{X}^{(Q)} = R_{(P\to Q)}\vec{X}^{(P)} + \vec{t}_{(P\to Q)}, \tag{2.1}$$

where $\vec{X}^{(P)}$ and $\vec{X}^{(Q)}$ denote the coordinates of an arbitrary 3D point $\vec{X} \in \mathbb{R}^3$ expressed in the corresponding (P) and (Q) frames, $R_{(P\to Q)} \in \mathbf{SO}(3)$ denotes the orthonormal rotation matrix, and $\vec{t}_{(P\to Q)}$ denotes the translation vector from the origin of (P) to the origin of (Q). The transformation has six degrees of freedom (DOF). It is fully parameterized by $\vec{t}_{(P\to Q)}$ and $R_{(P\to Q)}$. If such parameters are given or can be calculated, any point in one of the reference frame can be transformed to its coordinates in the other one. Fig. 2.1 illustrates the transformation by a simple example.

By specifying the 3D point in Equation (2.1) using two special points, i.e., the origins of the two coordinate systems $\vec{O}_P$ and $\vec{O}_Q$, the following relations can be obtained:

$$\vec{O}_P^{(Q)} = \vec{t}_{(P\to Q)}; \tag{2.2}$$

$$0 = R_{(P\to Q)}\vec{O}_Q^{(P)} + \vec{t}_{(P\to Q)}. \tag{2.3}$$

Furthermore, the inverse transformation can be derived:

$$R_{(Q\to P)} = R^{-1}_{(P\to Q)} = R^{T}_{(P\to Q)}; \tag{2.4}$$

$$\vec{t}_{(Q\to P)} = \vec{O}_Q^{(P)} = -R^{T}_{(P\to Q)}\vec{t}_{(P\to Q)}. \tag{2.5}$$

Optionally, the rigid body transformation between two reference frames can also be parameterized by elements of special Euclidean group $\mathbf{SE}(3)$. By choosing one coordinate system as a reference, e.g., (Q), it can be parameterized as an identity matrix $C_Q = I_4 \in \mathbf{SE}(3)$. Another coordinate frame can be represented by

$$C_P = C_{(P\to Q)} = \begin{bmatrix} R_{(P\to Q)} & \vec{t}_{(P\to Q)} \\ 0 & 1 \end{bmatrix} \in \mathbf{SE}(3). \tag{2.6}$$

An advantage of such parameterization is that if the point is represented by homogeneous coordinates, i.e.,

$$\tilde{X} = \begin{bmatrix} \vec{X} \\ 1 \end{bmatrix} \in \mathbb{P}^3, \text{ with } \quad \vec{X} \in \mathbb{R}^3, \tag{2.7}$$

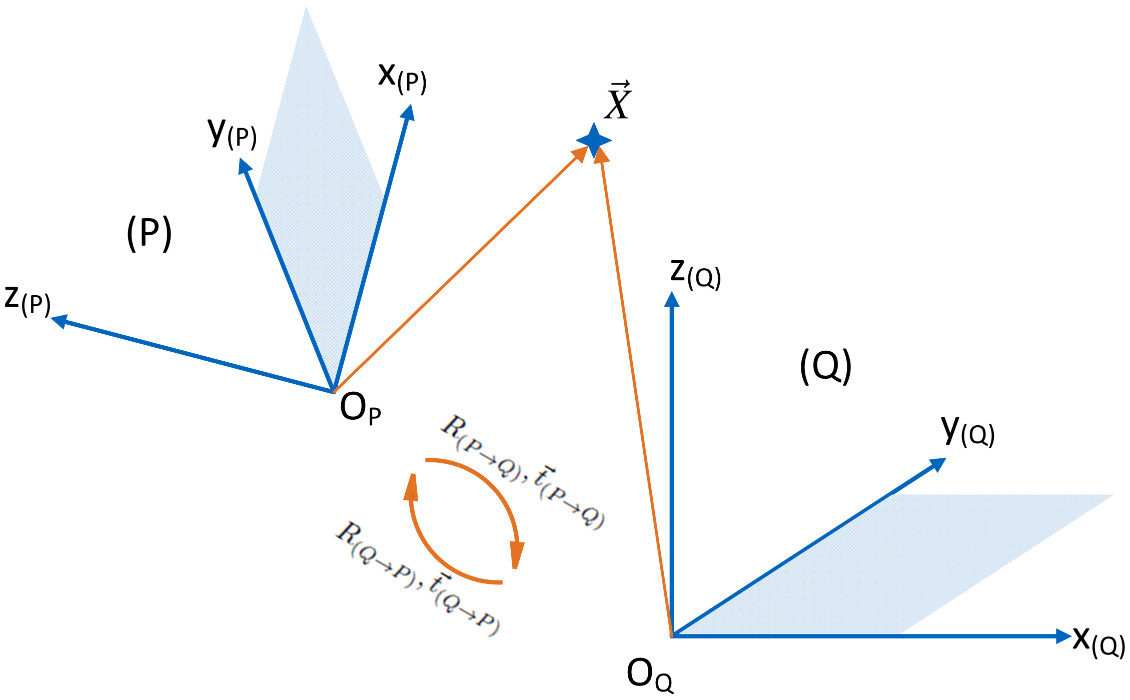


Figure 2.1: Basic coordinate transformation between two reference frames

the coordinates of the point can be easily transformed between two frames by

$$\tilde{X}^{(Q)} = C_{(P\to Q)}\tilde{X}^{(P)}, \tag{2.8}$$

where $\tilde{X}^{(P)}, \tilde{X}^{(Q)} \in \mathbb{P}^3$. In this representation, the inverse transformation is simply the inverse of the matrix:

$$C_{(Q\to P)} = \begin{bmatrix} R^T_{(P\to Q)} & -R^T_{(P\to Q)}\vec{t}_{(P\to Q)} \\ 0 & 1 \end{bmatrix} = C^{-1}_{(P\to Q)}. \tag{2.9}$$

Furthermore, consecutive transformation among multiple reference frames can be calculated by the multiplication of the transformation matrices, e.g.,

$$C_{(P\to Q)} = C_{(R\to Q)}C_{(P\to R)}. \tag{2.10}$$

The two ways of parameterizing the transformation between different coordinate systems are mathematically equivalent. One can choose either one according to the problem to simplify the expression in derivation.

2.2 Camera Model

The digital image sensor of cameras is the core sensing technology in visual navigation. A digital image sensor can generate free electrons from the sensed photons on the image plane, and create electrical signals according to the accumulated electrons during the exposure period. The signal is sampled at discrete pixels that are uniformly distributed in the image plane. As a result, through particular onboard signal processing such as filtering and quantization, the sensor generates a digital image I, which represents an amplitude measure of the illuminance during the exposure time on the image plane $\Omega \subset \mathbb{R}^2$. For a grayscale image, $I \in \mathbb{Z}^{N_h \times N_w}$ is an integer-valued matrix that represents the amount of intensity values at each pixel, where N_h and N_w are the number of the pixels in the dimensions of the height and the width respectively. In most common cases, intensity values range from 0 (black) to 255 (white) as the result of 8 bits quantization.

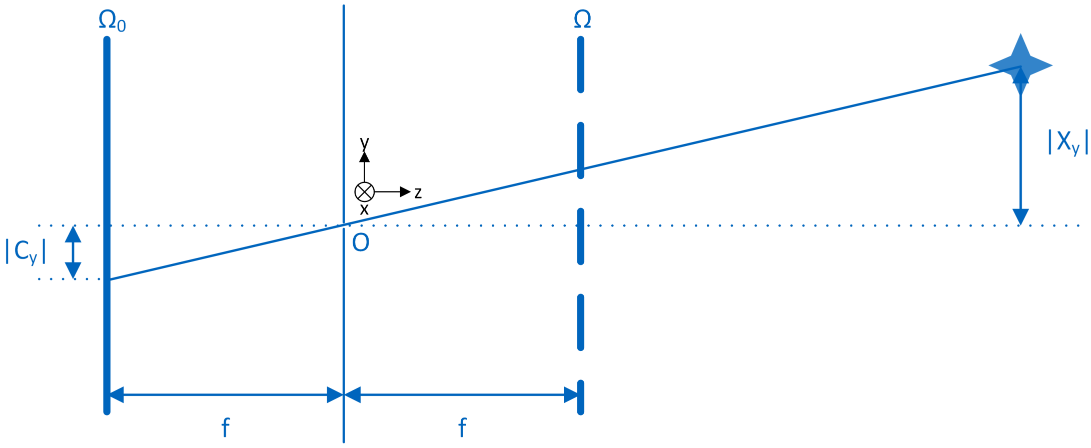


Figure 2.2: Pinhole camera model

The intensity values in the measurement image I are noisy due to various error sources. We model the measurement noise as additive noise:

$$I = I_0 + n_I. \tag{2.11}$$

The noise n_I is normally referred as photometric noise in literatures, since it represents the raw photometric error in the image intensity values.

Modern image sensors are capable of generating colored digital images with multiple channels of color space in order to adapt to the perception of human eyes. However, for autonomous visual navigation applications, it is essential to extract geometric information from the luminance of the images, and the processing should be ideally real-time. The gain in geometry extraction from the color information is usually insignificant compared to using monochrome luminance, while processing colored images requires at least three times as much the computational power as grayscale image with the same resolution. Moreover, with the same generation of technology, a monochrome camera can achieve higher frame rate than a colored camera with similar costs and resolution, which is crucial for reliable motion tracking using vision. As a result, in the following part of this work, it is assumed by default that the image measurements are in grayscale. The conversion from colored to grayscale image and the conversions among different color spaces can be found in manuals such as [13] and [14].

Most of the digital cameras are designed based on the notion of pinhole imaging, which captures the light rays passing through the aperture of the lens. Fig. 2.2 shows the projective geometry of the pinhole model. Assume the origin of the reference frame is at the center of the aperture. For a point in three-dimensional (3D) space with coordinates $\vec{X} = [X_x, X_y, X_z]^T$, the model approximates the projection of the point onto the camera's image plane Ω_0 as an ideal pinhole camera. The image plane is perpendicular to the optical axis and the distance between the plane and the aperture center is the focal length f. According to Fermat's principle in optics, the projection of the 3D point on the image plane has coordinates

$$\vec{c} = [c_x, c_y, c_z]^T = \left[-f\frac{X_x}{X_z}, -f\frac{X_y}{X_z}, -f \right]^T. \tag{2.12}$$

Concerning the projection using the pinhole model, the image of an object on the plane Ω_0 has been rotated by π around the z-axis compared to the original object. To simplify the analysis, one can introduce an equivalent virtual image plane Ω, which is at the same side of the lens as the visible object. Ω is parallel to Ω_0 and is located at distance f in front of the aperture. The same points projected on the two planes are bijectively related by a rotation mapping. As a result, the projection of an object on Ω will have the same size as the original image but is no longer rotated. Without loss of generality, Ω is used to refer to the image plane in the following parts of this dissertation.

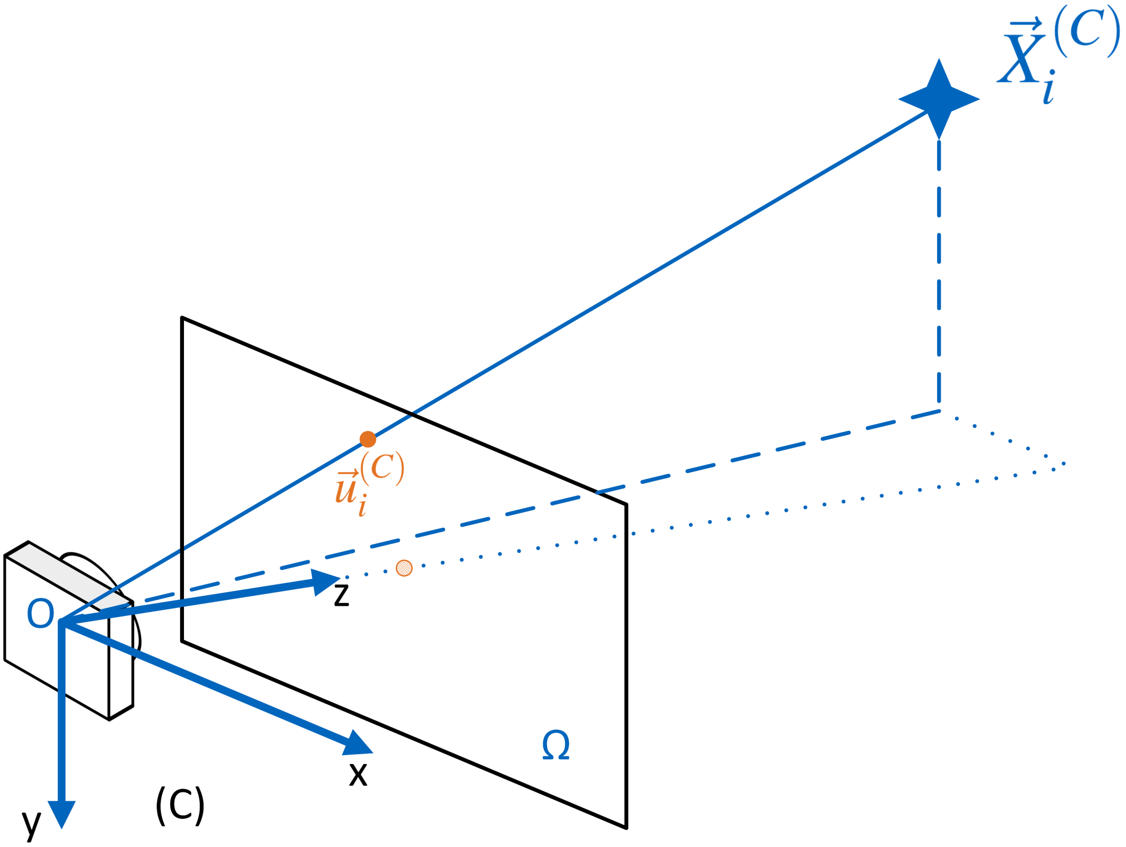


Figure 2.3: Camera reference frame

The camera reference frame (C) is defined as in Fig. 2.3. The origin of the reference frame is set at the projection center, i.e., the center of the lens aperture. For a 3D point $\vec{X}_i^{(C)} = [X_x^{(C)}, X_y^{(C)}, X_z^{(C)}]$, the projection of it on the image plane is

$$\vec{u}_i^{(C)} = \left[f\frac{X_x^{(C)}}{X_z^{(C)}}, f\frac{X_y^{(C)}}{X_z^{(C)}}, f \right]^T . \tag{2.13}$$

Following the convention of the pixel-counting, the two-dimensional (2D) position of a point in an image I is measured from the top-left corner of the image. As a result, the corresponding 2D position $u_i = [u_{i,x}, u_{i,y}]^T \in \Omega \subset \mathbb{R}^2$ can be related to the 3D point as

$$u_i = \left[f\frac{X_x^{(C)}}{X_z^{(C)}} + \frac{N_w}{2}, f\frac{X_y^{(C)}}{X_z^{(C)}} + \frac{N_h}{2} \right]^T . \tag{2.14}$$

Using homogeneous coordinates, the perspective projection can be written as the following linear equation

$$\tilde{u}_i = d_i \begin{bmatrix} u_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & N_w/2 \\ 0 & f & N_h/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_x^{(C)} \\ X_y^{(C)} \\ X_z^{(C)} \end{bmatrix} = K_C \vec{X}_i^{(C)}, \tag{2.15}$$

where $d_i = X_z^{(C)}$ is the depth of the point. The matrix K_C in Eqn. (2.15) is called camera intrinsic matrix. Its parameters are only dependent on the camera and the lens used. It should be mentioned that the linearity only holds for the homogeneous coordinate representation. In Euclidean coordinates, the mapping from the 3D point coordinates to the 2D position on the image plane is non-linear.

In practice, the optical system of a camera does not perfectly match the pinhole model. As a result, the obtained image has lens distortion to be modeled. For most of the cameras, the main geometric abberation of lenses is modeled as radial distortions and tangential distortions, which are reviewed in the state-of-the-art work by Fryer and Brown in [15]. Quantitatively, the undistorted 2D point u_i can be modeled as a function of distorted measurement $\breve{u}_i = [\breve{u}_{i,x}, \breve{u}_{i,y}]^T$ as

$$u_i = \breve{u}_i + \delta u_{ri} + \delta u_{ti} = \breve{u}_i + \begin{bmatrix} \delta x(k_1 r^2 + k_2 r^4 + k_3 r^6 + o(r^8)) \\ \delta y(k_1 r^2 + k_2 r^4 + k_3 r^6 + o(r^8)) \end{bmatrix} + \begin{bmatrix} 2p_1 \delta x \delta y + p_2(r^2 + 2\delta x^2) \\ p_1(r^2 + 2\delta y^2) + 2p_2 \delta x \delta y \end{bmatrix}, \tag{2.16}$$

where $\delta x = \breve{u}_{i,x} - N_w/2$, $\delta y = \breve{u}_{i,y} - N_h/2$ are the 2D coordinates with respect to the principal point, and $r = \sqrt{\delta x^2 + \delta y^2}$ is the corresponding radial distance. As an approximation, high order terms in the model are ignored, which have little impact on the accuracy of the model according to [15]. The radial distortion δu_{ri} refers to the barrel or pincushion distortion, which is symmetric with respect to the optical axis. The cause of it is the fact that the optical system cannot produce perfect rectilinear perspective projection. The tangential distortion occurs because the lenses are not perfectly parallel to the imaging plane. Additionally, in practice, the focal length of the x and y axis can be slightly different, and the two axes can be slightly skew due to imprecision in the manufacture. A more general expression of the camera intrinsic matrix can be written as

$$K_C = \begin{bmatrix} f_x & \gamma_{\text{skew}} & N_w/2 \\ 0 & f_y & N_h/2 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.17}$$

where γ_{skew} is the skew coefficient which is usually tiny for a well manufactured camera.

Using known visual patterns, e.g., a checkerboard, the lens distortion parameters $[k_1, k_2, k_3, p_1, p_2]$ as well as the intrinsic matrix K_C can be estimated by camera calibration methods such as [16] and [17]. Because the sensors are calibrated cameras in most applications of visual navigation, in the following part of this work, it is assumed that K_C is known and the lens distortion has already been corrected for the 2D points in the image, i.e., only the undistorted location u_i is considered.

The transformation between the camera inertial frame (C) and other reference frames, e.g., the world frame denoted by (W), is dependent on the position and attitude of the camera. The camera position is always set to be the projection center of the lens, i.e., point O in Fig. 2.3. For a camera at position $\vec{c}^{(W)} \in \mathbb{R}^3$ in the world frame, the following equation describes the transformation between the two frames:

$$\vec{X}^{(C)} = R_{(W\to C)}\vec{X}^{(W)} + \vec{t}_{(W\to C)}, \tag{2.18}$$

where $\vec{X}^{(W)} \in \mathbb{R}^3$ is the coordinates of an arbitrary point in world frame, and $\vec{X}^{(C)}$ is the same point in camera frame. $R_{(W\to C)}$ denotes the rotation between the two reference frames, and its inverse $R_{(C\to W)} = R^T_{(W\to C)} \in \mathbf{SO}(3)$ is used to represent the attitude of the camera in the world frame. $\vec{t}_{(W\to C)} = -R_{(W\to C)}\vec{c}^{(W)}$ is the translation between the two reference frames. As a result, the projective geometry of a 3D point in world frame is described by

$$\tilde{u}_i = d_i \begin{bmatrix} u_i \\ 1 \end{bmatrix} = K_C \begin{bmatrix} R_{(W\to C)} | -R_{(W\to C)}\vec{c}^{(W)} \end{bmatrix} \begin{bmatrix} \vec{X}_i^{(W)} \\ 1 \end{bmatrix} = P\tilde{X}_i^{(W)}, \tag{2.19}$$

where $\tilde{X}_i^{(W)} \in \mathbb{P}^3$ is the homogeneous coordinates of the point in world frame, and P is called projection matrix.

In Euclidean space, the 2D coordinates of a projected point in the image plane is related to the camera pose and the 3D location of the point by

$$u_i = \frac{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tilde{u}_i}{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tilde{u}_i} = \frac{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} K_C R^T_{(C\to W)}(\vec{X}_i^{(W)} - \vec{c}^{(W)})}{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} K_C R^T_{(C\to W)}(\vec{X}_i^{(W)} - \vec{c}^{(W)})}. \tag{2.20}$$

In order to simplify the notation, a function $\pi(\cdot) : (\mathbb{R}^3, \mathfrak{se}(3)) \to \mathbb{R}^2$ is defined to map the 3D point position and the extrinsic parameters, i.e., the camera pose using particular parameterizations, to the 2D coordinates on the image plane. The camera pose can be represented using Lie algebra of $\mathbf{SE}(3)$ group or other equivalent parameterizations. A brief introduction of parameterizing rotations and poses using Lie groups and Lie algebras can be found in [18]. Optionally, one can separate the parameterization of the camera pose to attitude and position. For instance, if a rotation matrix $R_{(C\to W)}$ is used to represent the camera attitude, the projection function in Eqn. (2.20) becomes

$$u_i = \pi(\vec{X}_i^{(W)}, R_{(C\to W)}, \vec{c}^{(W)}). \tag{2.21}$$

In visual navigation, the location of the 2D feature points are estimated using the noisy 2D image I in Eqn. (2.11). As a result, the 2D coordinates of the points are also imperfect. To distinguish from the photometric error in intensity domain, the error of the point location in the image plane $n_{ui} \in \mathbb{R}^2$ is normally referred as geometric error. The following measurement model is used in this work under Gaussian noise assumption

$$\mu_i = u_i + n_{ui}, \quad E\{n_{ui}\} = \vec{0}, \quad E\{n_{ui}n_{ui}^T\} = \Sigma_{ui}. \tag{2.22}$$

In practice, the geometric error is dependent on various ad-hoc factors, e.g., the type of feature detector, the property of the feature in the environment, etc. The Gaussian noise assumption may not always precisely model the geometric error distribution. However, developing a precise error model considering all the impacts is not part of the main goal of this thesis, and the feasibility of modelling the ad-hoc factors given restricted resources is disputable. Therefore, as a compromise, the visual positioning and sensor fusion methods developed in this work is based on the Gaussian assumption in Eqn. (2.22) for geometric noise.

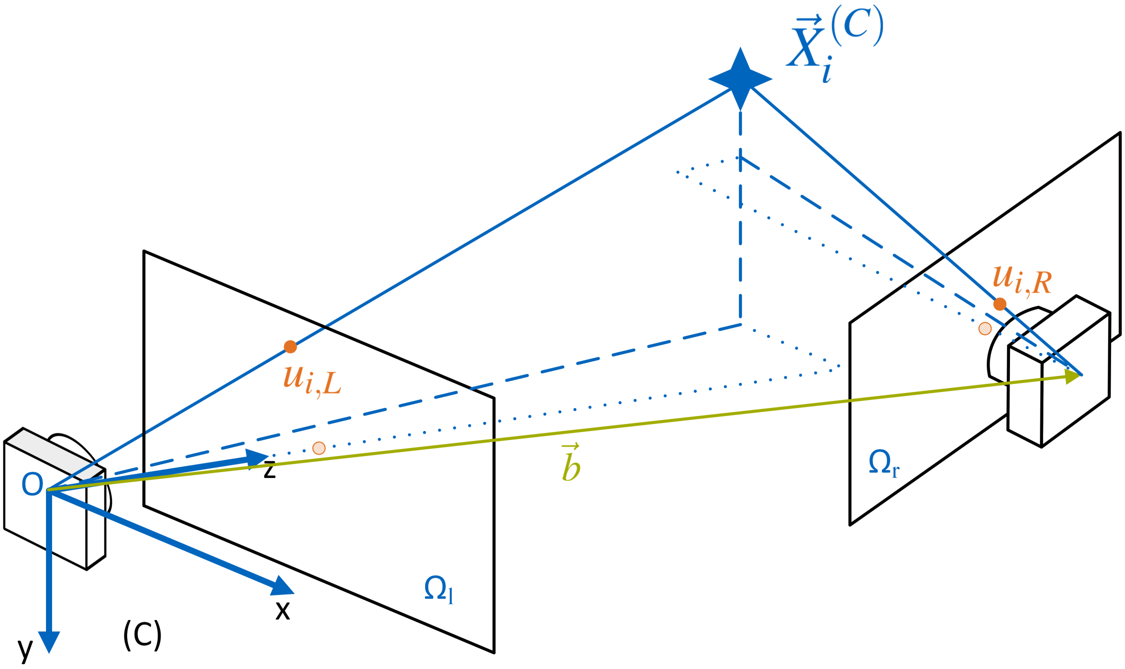


Figure 2.4: General model of a stereo camera rig.

2.3 Stereo Camera Rig Model

According to the pinhole camera model which was introduced in Section 2.2, the z-component of a point's coordinates in camera frame, i.e., the depth information, is unrecoverable from the 2D measurements on the image plane. However, using two cameras with known relative pose with common field of view, the depth of the matched feature points can be estimated. Therefore, stereo camera rigs (two cameras with a fixed baseline) are widely used in computer vision and robotics so that 3D information can be obtained.

Fig. 2.4 shows the general model of a stereo camera rig. For a stereo camera rig, the relative pose from the two cameras are fixed, which can be represented by a baseline vector $\vec{b}$ and a relative rotation $R_b \in \mathbf{SO}(3)$. The baseline vector and the relative rotation parameters are normally defined as extrinsic parameters of the stereo rig. Without loss of generality, it is assumed that the camera frame of the stereo rig (C) is the same as the camera inertial frame of the first camera. A feature point $\vec{X}_i^{(C)}$ is visible from both cameras. Applying pinhole model, Eqn. (2.23) describes the projection from the point to the first (left) camera's image plane $\Omega_L \subset \mathbb{R}^2$ as:

$$\tilde{u}_{i,L} = d_{i,L} \begin{bmatrix} u_{i,L} \\ 1 \end{bmatrix} = K_L \vec{X}_i^{(C)}. \tag{2.23}$$

At the same time, the projection of the point on the second (right) camera $\Omega_R \subset \mathbb{R}^2$ is expressed as:

$$\tilde{u}_{i,R} = d_{i,R} \begin{bmatrix} u_{i,R} \\ 1 \end{bmatrix} = K_R R_b (\vec{X}_i^{(C)} - \vec{b}). \tag{2.24}$$

Using the 2D measurements from both cameras $\mu_{i,L} = u_{i,L} + n_{uiL}$ and $\mu_{i,R} = u_{i,R} + n_{uiR}$ with independent noise processes for both cameras and noise covariance $E\{n_{uiL} n_{uiL}^T\} = \Sigma_{uiL}, E\{n_{uiR} n_{uiR}^T\} = \Sigma_{uiR}$ respectively, the 3D coordinates of the feature point can be estimated using triangulation by minimizing the reprojection error in both views:

$$\hat{X}_i(C) = \underset{\vec{X}_i^{(C)}}{\arg\min} \left\| \pi(\vec{X}_i^{(C)}, I_3, \vec{0}) - \mu_{i,L} \right\|^2_{\Sigma^{-1}_{uiL}} + \left\| \pi(\vec{X}_i^{(C)}, R_b, \vec{b}) - \mu_{i,R} \right\|^2_{\Sigma^{-1}_{uiR}}, \tag{2.25}$$

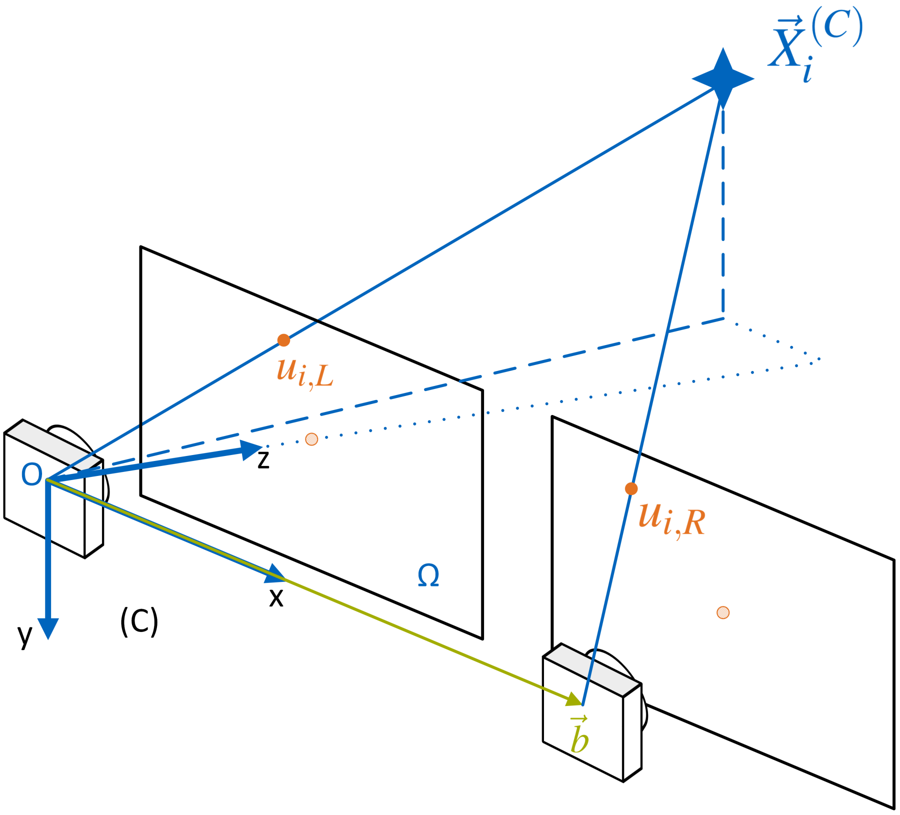


Figure 2.5: Projection model for a well-aligned stereo camera rig

where the projection function $\pi(\cdot)$ was defined in Section 2.2 and I_3 is three dimensional identity matrix. $\|\cdot\|_{\Sigma^{-1}}$ denotes the Mahalanobis distance with respect to covariance matrix Σ. As a direct solution of the nonlinear optimization in Eqn. (2.25), Hartley and Sturm proposed the state-of-the-art polynomial triangulation algorithm, which is shown to outperform most other triangulation algorithms. The details of the algorithm can be found in [19] and [20].

Using a pattern with known structure and size, e.g., a checker board, both the intrinsic parameters K_L, K_R and the extrinsic parameters $\vec{b}, R_b$ can be obtained through calibration. State of the art stereo camera calibration algorithms can be found in [21] and [22]. In this work, the cameras are calibrated using the camera calibration toolbox from OpenCV and Caltech, which is available at [23] and [24].

As a special case, if the two cameras in the stereo rig are the same, and are well aligned, i.e., $K_L = K_R = K_C, \vec{b} = [b_x, 0, 0]^T, R_b = I_3$ in Eqn. (2.25), the model is as expressed in Fig. 2.5. This special set-up has some nice properties. According to projective geometry, for a 2D feature point $u_{i,L} = [u_{ix,L}, u_{iy,L}]^T$, the possible positions of the corresponding 3D point $\vec{X}_i^{(C)}$ is represented by a line in the space. By projecting the line to the right view camera, one can obtain a line on the image plane, which is called epipolar line. The 2D measurement of the same point in the right image $u_{i,R}$ can only locate on the epipolar line. (A brief introduction of general epipolar geometry can be found in Section 4.1. More details can be found in [20].) For a well-aligned stereo camera rig, the epipolar lines are parallel to the x-axis of the camera frame. As a result, in the absence of noise, the y-coordinates of any pair of feature points are always the same. The search space of feature matching is reduced to searching along the pixels in the other image with the same y coordinates. (In practice, a margin is required in order to tolerate noisy measurements.) Moreover, the depth of the 3D point is the same for both views with coplanar image planes, i.e., $d_{i,L} = d_{i,R} = d_i$. Intuitively, it can be obtained from the geometry that

$$\frac{(b_x + u_{ix,R}) - u_{ix,L}}{b_x} = \frac{d_i - f}{d_i}. \tag{2.26}$$

Therefore, the depth can be calculated using the disparity of the x-component of the 2D feature locations. Denoting the disparity with $\delta x = u_{ix,L} - u_{ix,R}$, the depth can be recovered as

$$d_i = \frac{f}{\delta x} b_x. \tag{2.27}$$

Consequently, the coordinates of the 3D point can be reconstructed by

$$\vec{X}_i^{(C)} = K_C^{-1} d_i \begin{bmatrix} u_{ix,L} \\ u_{iy,L} \\ 1 \end{bmatrix} \tag{2.28}$$

In the presence of measurement noise, the disparity is calculated using the noisy 2D geometric measurements as $\hat{\delta x} = \mu_{ix,L} - \mu_{ix,R}$. The depth can be estimated using Eqn. (2.28) as

$$\hat{d}_i = \frac{f}{\mu_{ix,L} - \mu_{ix,R}} b_x, \tag{2.29}$$

which has some spatial uncertainty due to the noise in $\mu_{ix,L}$ and $\mu_{ix,R}$. If the left and right cameras are the same and are well aligned, it can be assumed that $\Sigma_{uiL} = \Sigma_{uiR}$.

The coordinates of the 3D point which minimizes the summation of the reprojection error in both views, as described in Eqn. (2.25), is estimated by

$$\hat{X}_i(C) = K_C^{-1} \hat{d}_i \begin{bmatrix} \mu_{ix,L} \\ (\mu_{iy,L} + \mu_{iy,R})/2 \\ 1 \end{bmatrix}. \tag{2.30}$$

The uncertainty of the estimated depth and 3D coordinates will be discussed in detail in Section 3.4.

In practice, for a camera with specific sensor size, the opening angle of the field of view is determined by the focal length of the main lens as

$$\varphi = 2 \tan^{-1} \left(\frac{N_w}{2f} \right). \tag{2.31}$$

The closest visible point in the common field of view has depth

$$d_{\min} = b_x \cot(\varphi/2) = 2 f b_x / N_w. \tag{2.32}$$

Fig. 2.6 illustrates an instance of the nearest visible point which locates in the common field of view of the left and right cameras.

By knowing the intrinsic and extrinsic parameters from calibration, the images obtained from a general stereo rig can be transformed to equivalent co-planar images using image rectification techniques, so that the 3D reconstruction is much easier using the model in Fig 2.5. A state-of-the-art image rectification algorithm was proposed in [25] by Loop and Zhang. Following the approach, images taken from a general stereo rig can be transformed and processed using the coplanar model.

For simplicity but without loss of generality, it is assumed in this dissertation that the image planes of two cameras are well aligned in a stereo rig so that the model in Fig. 2.5 applies, and the intrinsic matrices are the same for the left and right cameras, i.e., $K_L = K_R = K_C$.

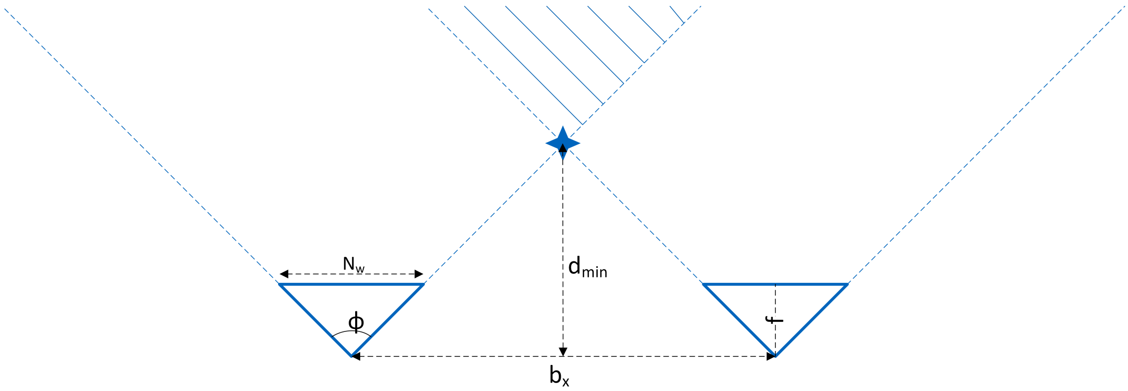


Figure 2.6: Nearest visible point in the common field of view.

2.4 Ranging Measurements

A rover is capable of measuring distance relative to a certain point using wireless radio signals. In a cooperative swarm, the radio links between vehicles can be utilized to obtain ranging measurements without using additional sensors. Fig. 2.7 shows the set-up for a one way ranging scenario. The transmitter (TX) sends a set of pilot signal $s(t)$ over the communication channel. Assuming the receiver (RX) has reliable channel estimation capability, the received signal after channel compensation can be denoted as $r(t) = s(t - \tau - \delta t) + n(t)$, where τ is the propagation delay, δt the clock offset between TX and RX, and $n(t)$ the noise. Accurate ranging measurements can be obtained using time of arrival (ToA) estimates if the clocks of the transmitter and the receiver are well synchronized, i.e. δt is known. By correlating the received signal with a local replica of the pilot signal, the ToA estimate can be obtained by

$$\hat{\tau}_{\text{ToA}} = \arg\max_{\tau_{\text{ToA}}} \int s^*(t - \tau_{\text{ToA}} - \delta t) r(t) dt. \tag{2.33}$$

Ideally, in noise-free case, the peak of the correlation will be found at $\tau_{\text{ToA}} = \tau$. Then, the ranging measurement based on ToA is calculated as $\rho = c_0 \hat{\tau}_{\text{ToA}}$ with c_0 the propagation speed of the wavefront. However, in many applications, the clock synchronization cannot fulfill the ranging accuracy requirements. In such situations, the ToA estimate $\hat{\tau}_{\text{ToA}}$ is biased from the true propagation delay τ by a clock offset, which can result in a large bias in the corresponding ranging measurement ρ, since c_0 is essentially the speed of the light in vacuum.

In order to mitigate the impact of the clock synchronization, it is chosen to utilize two way ranging based on round-trip delay (RTD) to measure the distance. Fig. 2.8 shows the concept of the RTD based ranging. A rover transmits pilot signals to another node, e.g., another rover or a base station. Then, the receiver side relays the pilot signal back. The rover receives the return link signal $r_{\text{RL}}(t) = s(t - \tau_{\text{FL}} - \tau_{\text{RL}}) + n_{\text{RTD}}(t)$, since the clock offsets $\pm\delta t$ cancel. The received signal is correlated with the local replica to estimate the

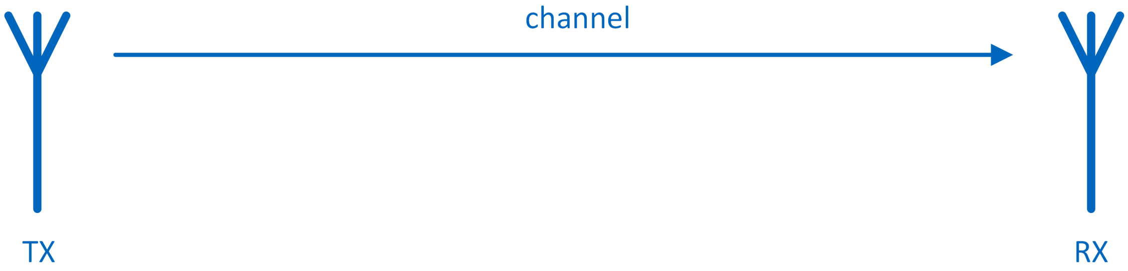


Figure 2.7: One way ranging based on ToA.

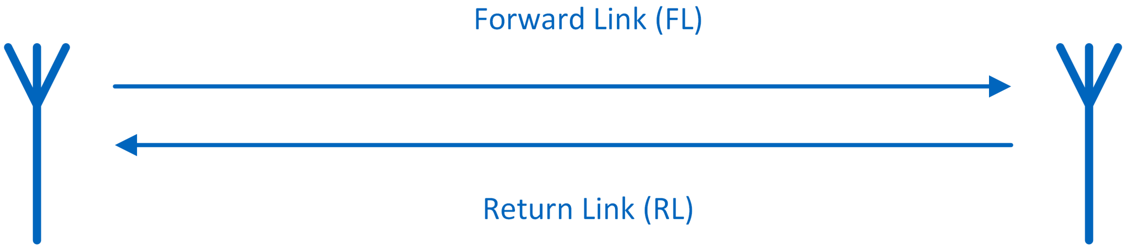


Figure 2.8: Two way ranging based on RTD.

round-trip delay as

$$\hat{\tau}_{\text{RTD}} = \arg\max_{\tau_{\text{RTD}}} \int_0^{\infty} s^*(t - \tau_{\text{RTD}}) r_{\text{RL}}(t) dt. \tag{2.34}$$

The ranging measurement is calculated as $\rho = c_0 \hat{\tau}_{\text{RTD}}/2$. Since the clock offsets in the forward link and the back link are cancelled out, the RTD based ranging approach does not require clock synchronization between devices. The clock offset impact can be ignored in RTD based methods, as long as the clocks on the platforms are sufficiently stable over a short period of time. The detailed system design of an RTD-based ranging system is introduced in [5].

For a transmitter located at $\vec{x}_t(t)$ and a receiver located at $\vec{x}_r(t)$, the ranging measurement model can be generally described as:

$$\rho(t) = F(t) + \eta(t), \tag{2.35}$$

where $F(t) = \|\vec{x}_t(t) - \vec{x}_r(t)\|$ is the true range between the two points at time instant t and $\eta(t)$ is the measurement noise. If the channel can be accurately estimated, the ranging noise can be modelled as a zero-mean Gaussian random variable, i.e.,

$$E\{\eta(t)\} = 0, \quad E\{\eta(t)\eta^T(t)\} = \Sigma_\rho(t). \tag{2.36}$$

Using the wireless radio based ranging sensor, the range is estimated with respect to the reference point of the antenna. When the ranging measurements are fused with the camera measurements, there is a fixed transformation between the two reference frames. For a mobile robotic platform, the relative pose between the camera and the antenna can be measured in calibration phase, i.e., $\vec{x}_a^{(C)}$ can be obtained from calibration. If the ranging antenna is mounted sufficiently close to the camera projection center (compared with the ranging noise level), the impact can be omitted, otherwise this offset must be corrected when fusing ranging measurements with images. For simplicity but without loss of generality, in the following part of this work, the mismatch between the two frames are assumed to be corrected using the calibration data, so that the corrected origin of the camera frame (C) aligns with the antenna reference point.

2.5 System Model of a Robotic Swarm

As argued in Chapter 1, robotic swarms have several advantages in efficiency and robustness in exploration tasks compared with a single Mars rover. A cooperative robotic swarm is a group of decentralized platforms which move individually but cooperate with each other through communication links. Each swarm element is capable of navigating autonomously using onboard sensors.

In this work, the vehicles are only equipped with monocular cameras and radio front-ends with ranging capability. Basic communications among swarm elements are available through the wireless radio links. Compared with multi-agent SLAM based on map merging on a central control unit such as [26, 27, 28], swarm based approach is more robust against single point failures. However, it requires distributed operations on each rover, so that the computational power are more restricted and the communication delay is less tolerated. As a result, in this thesis we are committed to develop swarm navigation solutions which do not require to transmit images or feature descriptors.

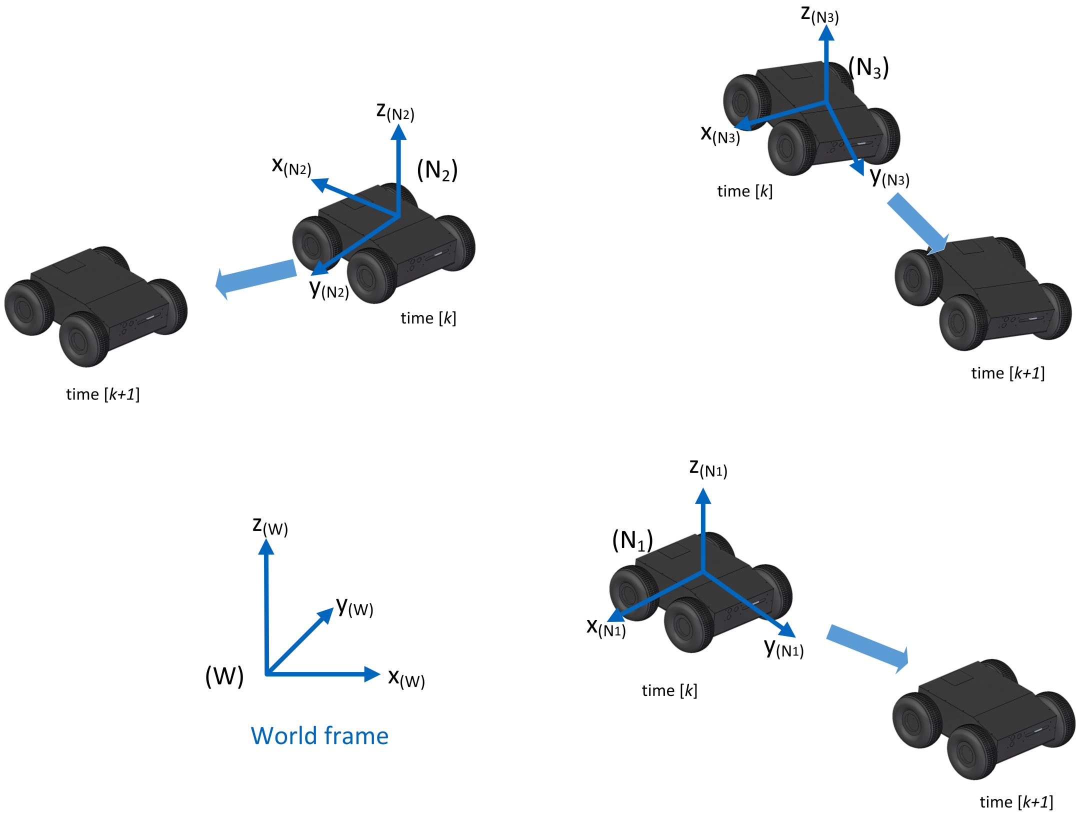


Figure 2.9: Reference frames in a robotic swarm.

Fig. 2.9 shows an instance of UGV (Unmanned Ground Vehicle) swarms. The motion of the rovers are constrained to be planar. Nevertheless, the feature points in the environment are distributed in 3D space. We define a navigation frame (N) as a fixed reference frame for each rover with its origin at the starting location of the rover. The navigation frame of each rover is related to the world reference frame by a specific transformation dependent on the initial position and attitude of vehicle. For rover j, any point represented in its navigation frame (N_j) can be transformed from the world frame into that frame by:

$$\vec{X}_i^{(N_j)} = R_{(W \to N_j)} \vec{X}_i^{(W)} + \vec{O}_W^{(N_j)}, \tag{2.37}$$

where $\vec{O}_W^{(N_j)}$ denotes the coordinates of the world frame origin in navigation frame (N_j). In planar motion, the poses (including position and attitude) of the agents in a robotic swarm can be illustrated as in Fig. 2.10. In the plot, the long edge of the triangles illustrates the virtual image plane of the onboard cameras. Orange triangles indicate the poses of the rovers at the current time instant, while the blue ones indicate the past trajectories.

Moreover, $(^{\mathsf{j}}k)$ is used to express the j-th rover's local reference frame at time k, which varies as the rover moves. Assuming the camera is fixed on the rover, the local body frame can be transformed to the local camera frame by a fixed transformation. Without loss of generality, it is assumed that the camera is mounted on the rover to look forward, so that $(^{\mathsf{j}}k)$ aligns with the local camera frame. (The detailed definition of the camera frame is described in section 2.2.) For other set-ups, the relative transformation can be obtained from hardware calibration. The relations of the world frame, navigation frame and the local camera frame are illustrated in Fig. 2.11.

Let $^{\mathsf{j}}\vec{c}_{[k]}^{(W)} \in \mathbb{R}^3$ be the position of the robot j in world frame (W) at time k, and $R_{(^{\mathsf{j}}k \to W)} \in \mathbf{SO}(3)$ be the rotation matrix representing the absolute attitude in the world frame. There are various ways to parameterize the orthonormal rotation matrix with 3 degrees of freedom, such as using Euler angles, unit

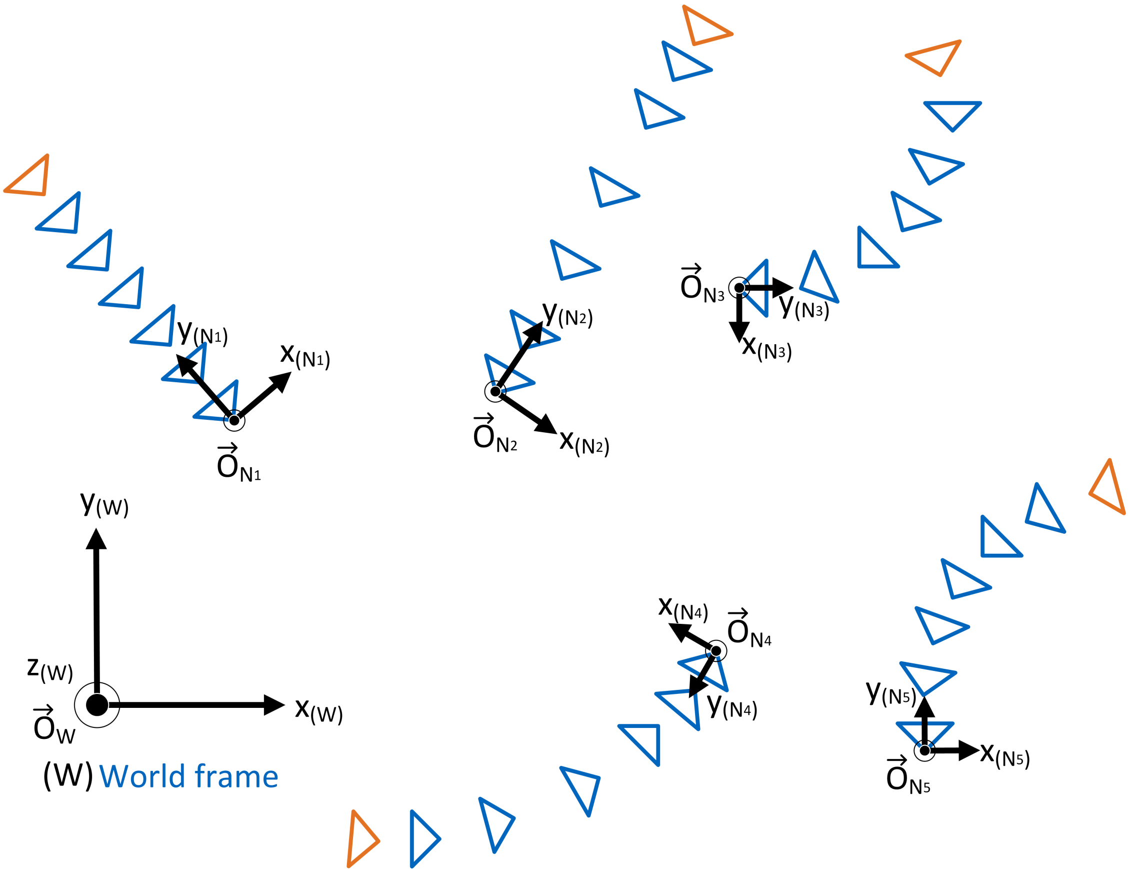


Figure 2.10: Reference frames in a robotic swarm in planar motion.

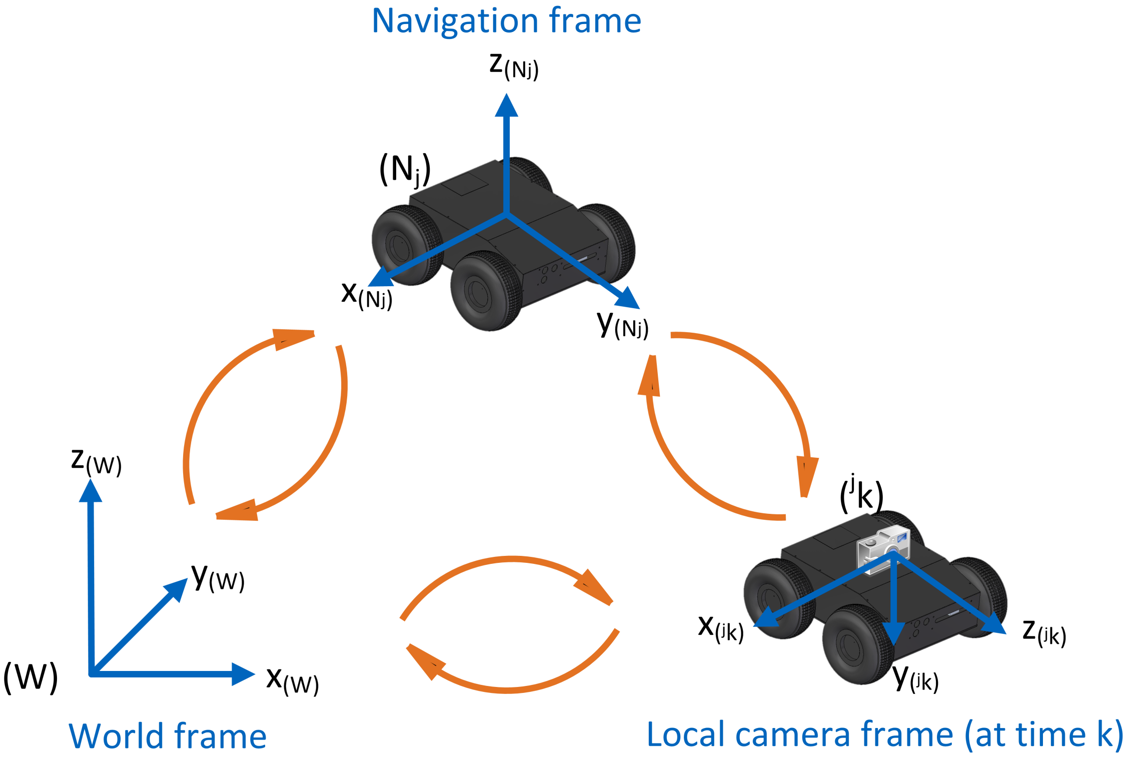


Figure 2.11: Reference frames of the swarm element j.

quaternions, Rodrigues parameters, and Lie algebras, etc. A detailed comparison among different attitude parameterizations can be found in [29]. Generally, they can be transformed to each other. Assuming that the rovers and the mounted onboard sensors are rigid, the pose of rover j at time instant k can be expressed by an element of special Euclidean group as

$$^{\mathrm{j}}C_{[k]}^{(W)} = \begin{bmatrix} R_{(\mathrm{j}k\to W)} & ^{\mathrm{j}}\vec{c}_{[k]}^{(W)} \\ 0 & 1 \end{bmatrix} \in \mathbf{SE}(3). \tag{2.38}$$

For vehicles constrained to be moving in a plane, the world reference frame (W) can be selected so that the $z-$axis is perpendicular to the motion plane, i.e.,

$$\forall j, k, \quad ^{\mathrm{j}}\vec{c}_{[k]}^{(W)} = \begin{bmatrix} ^{\mathrm{j}}\vec{\beta}_{[k]}^{(W)} \\ 0 \end{bmatrix} \in \mathbb{R}^3, \tag{2.39}$$

where

$$^{\mathrm{j}}\vec{\beta}_{[k]}^{(W)} = \begin{bmatrix} ^{\mathrm{j}}c_{[k],x}^{(W)}, {}^{\mathrm{j}}c_{[k],y}^{(W)} \end{bmatrix}^T \in \mathbb{R}^2 \tag{2.40}$$

denotes the 2D position in the motion plane. Moreover, the attitude of the rover can be parameterized by a one dimensional heading angle $^{\mathrm{j}}\phi_{[k]}^{(W)}$ in the plane. Consequently, the pose of vehicle j at time instant k can be parameterized by three parameters as

$$^{\mathrm{j}}x_{[k]}^{(W)} = \begin{bmatrix} ^{\mathrm{j}}c_{[k],x}^{(W)}, {}^{\mathrm{j}}c_{[k],y}^{(W)}, {}^{\mathrm{j}}\phi_{[k]}^{(W)} \end{bmatrix}^T \in \mathbb{R}^3. \tag{2.41}$$

The position and attitude of the rover in 3D space is calculated from the three pose parameters of $^{\mathrm{j}}x_{[k]}^{(W)}$ by

$$^{\mathrm{j}}\vec{c}_{[k]}^{(W)} = \begin{bmatrix} ^{\mathrm{j}}c_{[k],x}^{(W)} \\ ^{\mathrm{j}}c_{[k],y}^{(W)} \\ 0 \end{bmatrix}, \quad R_{(\mathrm{j}k\to W)} = \begin{bmatrix} \cos(^{\mathrm{j}}\phi_{[k]}^{(W)}) & 0 & \sin(^{\mathrm{j}}\phi_{[k]}^{(W)}) \\ -\sin(^{\mathrm{j}}\phi_{[k]}^{(W)}) & 0 & \cos(^{\mathrm{j}}\phi_{[k]}^{(W)}) \\ 0 & -1 & 0 \end{bmatrix}. \tag{2.42}$$

It is intuitive that the planar rigid body motion can be expressed using two-dimensional special Euclidean group $\mathbf{SE}(2)$. The poses parameterization is actually the vector form of the corresponding Lie algebra, i.e., $^{\mathrm{j}}x_{[k]}^{(W)} \in \mathrm{vec}(\mathfrak{se}(2))$. The reason for still using $\mathbf{SE}(3)$ to denote the rover poses is that even though the motion is constrained to be planar, the VSLAM problem still need to be able to cope with map points in 3D space. The model allows the future extension of the proposed methods to 3D SLAM.

The pose change of a dynamic rover follows the rigid body motion transformation:

$$^{\mathrm{j}}C_{[k+1]}^{(W)} = {}^{\mathrm{j}}C_{[k]}^{(W)} C_{(\mathrm{j}k+1\to \mathrm{j}k)}, \tag{2.43}$$

with

$$C_{(\mathrm{j}k+1\to \mathrm{j}k)} = \begin{bmatrix} R_{(\mathrm{j}k+1\to \mathrm{j}k)} & ^{\mathrm{j}}\vec{c}_{[k+1]}^{(\mathrm{j}k)} \\ 0 & 1 \end{bmatrix} \in \mathbf{SE}(3). \tag{2.44}$$

More specifically, with planar motion constraint, the rigid body transformation has only three degrees of freedom (DOF), which can be expressed by the 2D relative position $^{\mathrm{j}}\vec{\beta}_{[k+1]}^{(\mathrm{j}k)}$ and heading angle $^{\mathrm{j}}\phi_{[k+1]}^{(\mathrm{j}k)} = {}^{\mathrm{j}}\phi_{[k+1]}^{(W)} - {}^{\mathrm{j}}\phi_{[k]}^{(W)}$ as

$$^{\mathrm{j}}\vec{c}_{[k+1]}^{(\mathrm{j}k)} = \begin{bmatrix} ^{\mathrm{j}}\vec{\beta}_{[k+1]}^{(\mathrm{j}k)} \\ 0 \end{bmatrix}, \text{ with } \quad ^{\mathrm{j}}\vec{\beta}_{[k+1]}^{(\mathrm{j}k)} \in \mathbb{R}^2, \tag{2.45}$$

$$R_{(\mathsf{j}k+1\rightarrow\mathsf{j}k)}=\begin{bmatrix}\cos({}^{\mathsf{j}}\phi_{[k+1]}^{(\mathsf{j}k)}) & 0 & \sin({}^{\mathsf{j}}\phi_{[k+1]}^{(\mathsf{j}k)})\\ -\sin({}^{\mathsf{j}}\phi_{[k+1]}^{(\mathsf{j}k)}) & 0 & \cos({}^{\mathsf{j}}\phi_{[k+1]}^{(\mathsf{j}k)})\\ 0 & -1 & 0\end{bmatrix}. \tag{2.46}$$

As a result, for a robotic swarm consisting of N_r rovers, the poses of the rovers in world frame at time instant k can be represented as $\{{}^{\mathsf{j}}C_{[k]}^{(W)}|j=1...N_r\}$. In planar motion case, the poses can be parameterized by $\{{}^{\mathsf{j}}x_{[k]}^{(W)}|j=1...N_r\}$. The estimation of the rovers' trajectories until current instant N_k, i.e., estimating $\{{}^{\mathsf{j}}x_{[k]}^{(W)}|j=1...N_r, k=1...N_k\}$ using the onboard cameras and the intra-swarm radio links, is the core problem of swarm navigation as well as the main goal of this dissertation.

3. Stand-alone Visual Navigation of a Single Vehicle – a Review and an Uncertainty Model

An autonomous vehicle can estimate its ego-motion based on visual cues. At the same time, a map of the environment can be created simultaneously using cameras. An overview of the estimation problem behind visual navigation techniques is provided in this chapter. In addition, the general methodology framework of visual navigation is introduced. The introduction is based on a stereo camera rig, since it is capable of recovering the depth information, which is lost in perspective projection, without using other sensors. On the contrary, the estimated vehicle trajectory and map point locations based on monocular vision is affected by a metric ambiguity, which is widely investigated by the research community as the scale problem. Subsequently, a general uncertainty model for visual SLAM is discussed based on the Cramér-Rao lower bound (CRLB) of the applied estimator. It is shown that from the aspect of estimation uncertainty, the advantage of a stereo rig over a monocular camera becomes increasingly marginal if the visual features are getting farther away while the stereo baseline length is limited. As a result, for miniaturized swarm platforms with limited size and costs, using a monocular camera and recovering the scale with other sensors can be a more favored solution than conventional stereo-vision based solution.

3.1 Framework of Visual Navigation using Cameras

For a single autonomous vehicle equipped with fix-mounted cameras, the pose of the vehicle and that of the camera are not distinguished in this dissertation, because the difference can be obtained by calibration. The images captured by a camera will change as the vehicle moves. By tracking particular image features associated with the static environment, methods can be applied to estimate the motion of the camera. Consequently, the relative pose, i.e., relative position and attitude, with respect to the initial position of the vehicle can be estimated using visual cues. Therefore, given a known initial position and attitude of the vehicle in a global coordinate frame, the following poses in the defined frame can be resolved. In this dissertation, we assume that the dynamic features in the environment can be detected and excluded using techniques such as outlier rejection, so that all the tracked features are static.

As introduced in section 2.2, the raw measurements obtained from a camera are pixel's intensities of the images. We denote the intensities of the image at time epoch k by $I_{[k]} \in \mathbb{Z}^{N_h \times N_w}$, where N_h and N_w are respectively the height and the width of the image in pixels, determined by the resolution of the sensor. In order to exploit the luminance measurements to estimate the geometric information, there are generally two categories of approaches:direct methods and indirect methods. Engel, Koltun and Cremers have reviewed different categories of approaches in [30].

The direct methods utilize the pixel intensity values directly to estimate the camera motion. It is assumed that the points corresponding to the visible pixels forms continuous surfaces in the 3D space and the luminance of them are invariant over short time. The direct approaches do not detect points of interest from the images, but monitor the luminance change of the points corresponding to the pixels. For each pixel, direct methods treat the intensity value as a one-dimensional photometric measurement. For a pixel at h-th row and w-th column of the image with intensity value $I_{[k]}(h, w)$ at time k, the position of the 3D map point corresponding to the pixel can be parameterized with one degree of freedom depth $d_{h,w}$ as:

$$\vec{X}_{h,w} = d_{h,w} K_C^{-1} \begin{bmatrix} w \\ h \\ 1 \end{bmatrix}, \tag{3.1}$$

where K_C is the camera intrinsic matrix. When the camera is moving, it is assumed that the pixel projected from the point $\vec{X}_{h,w}$ should also has intensity value $I_{[k]}(h,w)$. As a result, the motion of the camera is reflected by the variation of the intensity distribution over frames. The optimal estimate can be obtained by minimizing the photometric residual between the map points and the intensity measurements as:

$$\hat{x}_{[k+1]} = \underset{x_{[k+1]}}{\arg\min} \sum_{h=1}^{N_h} \sum_{w=1}^{N_w} \left\| I_{[k]}(h,w) - I_{[k+1]}\left(\pi(\vec{X}_{h,w}, x_{[k+1]})\right) \right\|^2_{\Sigma_I^{-1}}, \tag{3.2}$$

where $\pi(\vec{X}_{h,w}, x_{[k+1]}) : \mathbb{R}^3 \times \mathbf{se}(3) \rightarrow \mathbb{Z}^2$ denotes the function projects 3D point $\vec{X}_{h,w}$ to (rounded) pixel location in the image taken from the camera with pose $x_{[k+1]}$.

Direct methods can build dense maps by assigning each pixel to a map point in the 3D space, such as in work [31]. However, it should be mentioned that direct methods do not necessarily estimate the camera egomotion and 3D map points location in a dense sense using all pixels. When a dense map is not required, direct methods can only use the pixels with intensity gradients to obtain more accurate result with a sparse or semi-dense map, e.g., see the work [32, 33, 34] from Engel et al. Direct approaches can exploit various forms of information in the image without distinguishing it to be an edge, a corner, or other features. However, due to the basic assumptions, the direct methods are significantly sensitive to the light condition changes, shadows and occlusions. Additionally, the motion tracking directly using intensities has a nonlinear cost function (see Eqn. (3.2)) with many local minima. Initializing with the last image, the optimization solver requires a short baseline between two consecutive frames for a good initialization in order to ensure convergence to the correct solution.

On the other hand, the indirect methods first apply feature detectors to locate features of interests in the image, e.g., corners, edges, light blobs, etc., so that a set of geometric measurements $\{\mu_{i[k]} = S(I_{[k]}) \subset \Omega | i = 1, ..., N_p\}$ are established from the image intensities, in which $S(\cdot)$ is the function for feature location detection and $\Omega \subset \mathbb{R}^2$ denotes the 2D image plane. An indirect methods must determine the type of feature to detect. Edge detectors can well handle indoor artificial environments with particular a priori model. Nevertheless, for outdoor environments such us Mars exploration scenarios, stable corner features are most precious and widely used in feature-based visual navigation. For corner features, the extracted points location are used as two-dimensional geometric measurements for motion tracking, i.e., $\mu_{i,[k+1]} \in \mathbb{R}^2$. Moreover, the descriptive characteristics of the detected features, e.g., the local histogram, the local intensity gradients, etc., are extracted by a vector named as feature descriptor. Then the corner points are tracked in consecutive image frames by applying the feature descriptor, and the motion of the camera can be estimated by minimizing the residual between the projection of the tracked features and the geometric measurements:

$$\hat{x}_{[k+1]} = \underset{x_{[k+1]}}{\arg\min} \sum_{i=1}^{N_p} \left\| \mu_{i,[k+1]} - \pi(\vec{X}_i, x_{[k+1]}) \right\|^2_{\Sigma^{-1}_{u,i[k+1]}}, \tag{3.3}$$

where $x_{[k+1]}$ denotes the camera pose at time $k+1$. $\mu_{i,[k+1]}$ denotes the 2D geometric measurements of the i-th tracked feature with noise covariance $\Sigma_{u,i[k+1]}$, and X_i is the coordinates of the corresponding feature position in the 3D physical space.

It has been proved that five non-colinear points are theoretically sufficient for motion estimation between two images [35], and it is also verified that the number of tracked features is one of the most significant factors affecting the motion estimation accuracy [36]. Compared with direct methods, the extracted feature points are more stable to be tracked and located, since a unique multi-dimensional descriptor vector is assigned to each feature to characterize it. As an analog, the direct methods 'detect' the pixels as features, and build a one-dimensional feature descriptor by only using the intensity value, which is less representative than descriptors in indirect methods. Moreover, many corner detection methods such as [37] can achieve sub-pixel precision, while the direct methods are limited by the spatial sampling frequency of the pixels. Therefore, considering our visual navigation application scenario on Mars, we choose to use indirect methods based on reliable sparse feature points.

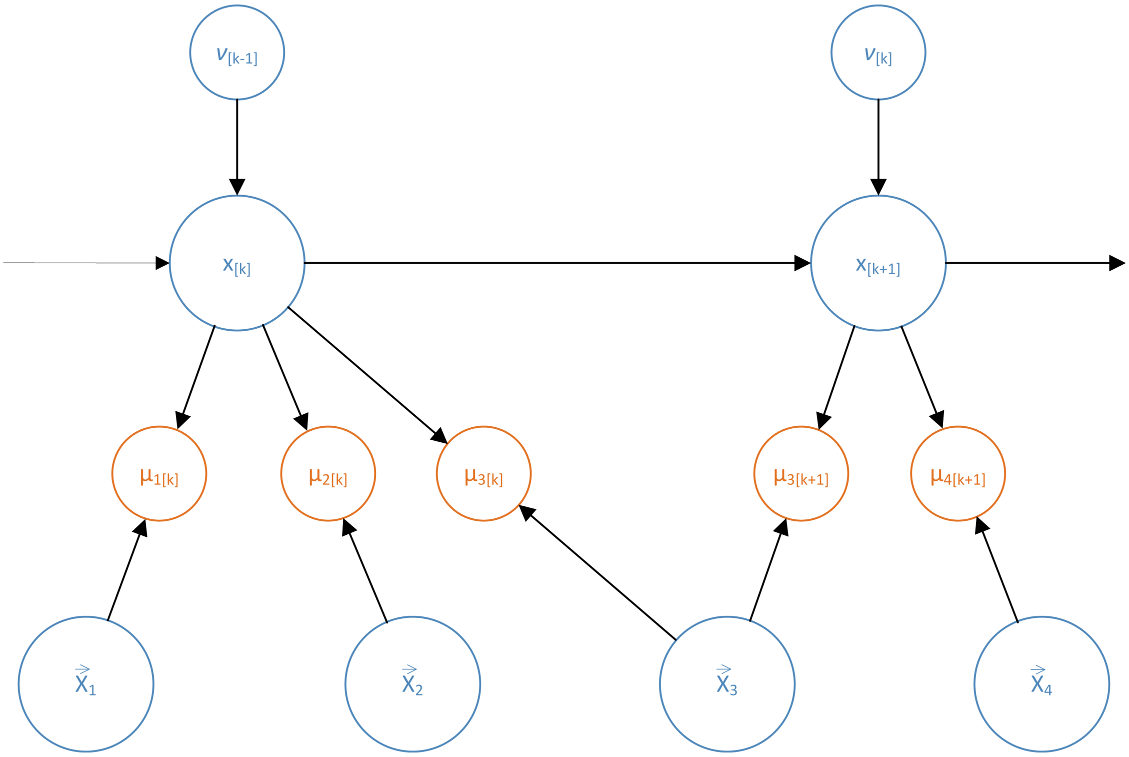


Figure 3.1: Bayesian network expression of a general instance of visual navigation

Denoting the camera pose at time k by a state vector $x_{[k]}$, Fig. 3.1 demonstrates a general instance of visual navigation scenarios for two consecutive time epochs using a Bayesian network model. We use the set $\{\vec{X}_i|i = 1, ..., N_p\}$ to denote the static 3D feature points in the surrounding environment. For the instance in the figure, $N_p = 4$. The visible features at time instant k is a subset of it, denoted by $\{\vec{X}_i|v_{i[k]} = 1\} \subseteq \{\vec{X}_i|i = 1, ..., N_p\}$, where $v_{i[k]}$ is a binary visibility masking that $v_{i[k]} = 1$ if feature i is visible to the camera at time instant k, otherwise $v_{i[k]} = 0$. In the example, $v_{1[k]} = v_{2[k]} = v_{3[k]} = v_{3[k+1]} = v_{4[k+1]} = 1$. The 2D position of a point $\vec{X}_i \in \mathbb{R}^3$ projected on the virtual image plane of camera k can be expressed as $u_{i[k]}$ using the camera model introduced in Section 2.2. It should be mentioned that the graph model can be adapted to both the monocular camera case and the stereo camera case. For a monocular camera VSLAM problem, $u_{i[k]} \in \mathbb{R}^2$, while for a stereo rig, the geometric measurement $u_{i[k]} = [u_{ix[k],L}, u_{iy[k],L}, u_{ix[k],R}, u_{iy[k],R}]^T \in \mathbb{R}^4$ includes measurements from both left and right cameras. $\mu_{i[k]}$ is the corresponding noisy measurement. $\nu_{[k]}$ denotes the control input at time k.

In a Bayesian network model, each random variable is conditionally independent of all its non-descendant nodes given all its parent nodes. Without loss of generality, the following assumptions are made:

1) The measurements of different features in the image plane are independent;
2) The state propagation of the vehicle is a Markov process, i.e., the parameters at time instant $k + 1$ is independent of earlier states given the state vector and the control input at time k;
3) If no further information or constraint is provided, the 3D positions of the feature points in the space are static and independent to each other.

As a result, the visual navigation problem at time instant N_k can be modeled by a directed graph as the Bayesian network in Fig. 3.1 (extended to time N_k). The joint probability density function of the random variables in the whole graph is

$$p_{joint} = p(x_{[0]}, x_{[1]}, ..., x_{[N_k]}, \nu_{[0]}, ..., \nu_{[N_k-1]}, \vec{X}_1, ..., \vec{X}_{N_p}, \mu_{1[1]}, ..., \mu_{N_p[1]}, ..., \mu_{N_p[N_k]}), \tag{3.4}$$

where $x_{[0]}$ is the state vector corresponding to the initial pose of the camera.

According to the conditional independence of the variables in the Bayesian network, the joint probability can be factorized as

$$p_{joint} = \prod_{i=1}^{N_p} \prod_{k=1}^{N_k} p(\mu_{i[k]}|\vec{X}_i, x_{[k]})^{v_{i[k]}} p(\vec{X}_i) p(x_{[k]}|x_{[k-1]}, \nu_{[k-1]}) p(\nu_{[k-1]}) p(x_{[0]}), \tag{3.5}$$

where $v_{i[k]}$ is the binary visibility masking for feature i at time k.

There are different applications with distinct problem set-ups in visual navigation. If the vehicle has no clue about its egomotion nor the surrounding environment, it is required to use cameras to estimate its motion while building a map simultaneously. This is known as the classic simultaneously localization and mapping (SLAM) problem in robotics. The problem can be formulated as a maximum a posteriori (MAP) estimation as

$$\{\hat{x}_{[k]}, \hat{X}_i\} = \arg\max_{\{x_{[k]}, \vec{X}_i\}} p(x_{[0]}, x_{[1]}, ..., x_{[N_k]}, \vec{X}_1, ...\vec{X}_{N_p} | \mu_{1[1]}, ..., \mu_{N_p[N_k]}, \nu_{[0]}, ..., \nu_{[N_k-1]}). \tag{3.6}$$

In relative positioning tasks, the a priori distribution of the initial states $x_{[0]}$ is assumed to be known. Without loss of generality, the initial position is assumed to be at $\vec{0}$ in the navigation coordinates frame. Moreover, the control input command is known to the platform, while it cannot obtain the a priori distribution of the feature points position $p(\vec{X}_i)$ in exploration missions. As a result, using the factorization in Eqn. (3.5), the optimal solution of the MAP estimator is:

$$\{\hat{x}_{[k]}, \hat{X}_i\} = \arg\max_{\{x_{[k]}, \vec{X}_i\}} \prod_{i=1}^{N_p} \prod_{k=1}^{N_k} p(\mu_{i[k]}|\vec{X}_i, x_{[k]})^{v_{i[k]}} p(x_{[k]}|x_{[k-1]}, \nu_{[k-1]}). \tag{3.7}$$

Therefore, the pose of the vehicle in visual SLAM tasks can be estimated by the following maximum likelihood (ML) estimator

$$\{\hat{x}_{[k]}, \hat{X}_i\} = \arg\max_{\{x_{[k]}, \vec{X}_i\}} \sum_{i=1}^{N_p} \sum_{k=1}^{N_k} \left(v_{i[k]} \log p(\mu_{i[k]}|\vec{X}_i, x_{[k]}) + \log p(x_{[k]}|x_{[k-1]}, \nu_{[k-1]}) \right). \tag{3.8}$$

The state transition of the platform can be modeled as:

$$x_{[k]} = T_{[k]}(x_{[k-1]}, \nu_{[k-1]}) + w_{[k]}, \tag{3.9}$$

where $T_{[k]}(x_{[k-1]}, \nu_{[k-1]})$ is the state transition function of the platform from time $k-1$ to time k, and $w_{[k]}$ is the process noise with covariance matrix Σ_w. With a precise motion model, the term $p(x_{[k]}|x_{[k-1]}, \nu_{[k-1]})$ in Eqn. (3.8) is determined by the process noise distribution. If a reliable motion model is unavailable, the covariance of the process noise would be large. Even though, the robustness of the solution is improved by including the motion model, because the estimation drift is still limited by the model when the measurements data has significantly poor quality, e.g., when the images suddenly blackout.

If the 2D geometric measurement noise distribution is modeled as Gaussian, the likelihood function writes

$$p(\mu_{i[k]}|\vec{X}_i, x_{[k]}) = \frac{1}{2\pi} |\Sigma_u|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mu_{i[k]} - u_{i[k]})^T \Sigma_u^{-1} (\mu_{i[k]} - u_{i[k]})}, \tag{3.10}$$

where $u_{i[k]} = \pi(\vec{X}_i, x_{[k]})$ is the projected 2D point locations given the parameters, and Σ_u is the covariance matrix of the measurements. The details of the projection function were introduced in Section 2.2. With the Gaussian noise assumption on both the measurement noise and the process noise, the ML estimator has the same solution as the following least-squares estimator:

$$\{\hat{x}_{[k]}, \hat{X}_i\} = \arg\min_{\{x_{[k]}, \vec{X}_i\}} \sum_{i=1}^{N_p} \sum_{k=1}^{N_k} \Big(v_{i[k]} (\mu_{i[k]} - \pi(\vec{X}_i, x_{[k]}))^T \Sigma_u^{-1} (\mu_{i[k]} - \pi(\vec{X}_i, x_{[k]})) \tag{3.11}$$

$$+ (x_{[k]} - T_{[k]}(x_{[k-1]}, \nu_{[k-1]}))^T \Sigma_w^{-1} (x_{[k]} - T_{[k]}(x_{[k-1]}, \nu_{[k-1]})) \Big). \tag{3.12}$$

In many applications of visual SLAM, a reliable motion model is difficult to achieve, e.g., for a handhold camera. In such cases, the noise covariance $\det(\Sigma_w) \gg \det(\Sigma_u)$. As a result, if the camera measurements are reliable while a precise motion model is missing, we can solve the problem with the measurements only:

$$\{\hat{x}_{[k]}, \hat{X}_i\} = \arg\min_{\{x_{[k]}, \vec{X}_i\}} \sum_{i=1}^{N_p} \sum_{k=1}^{N_k} v_{i[k]} (\mu_{i[k]} - \pi(\vec{X}_i, x_{[k]}))^T \Sigma_u^{-1} (\mu_{i[k]} - \pi(\vec{X}_i, x_{[k]})). \tag{3.13}$$

Eqn. (3.13) is the state-of-the-art bundle adjustment problem in computer vision.

The cost function in the optimization is non-linear with respect to the parameters $\{x_{[k]}, \vec{X}_i\}$ to be estimated. Therefore, in order to obtain the correct global minimum solution, it is essential to have good initial guess of the parameters. This requirement is fulfilled by applying motion tracking and map points reconstruction methods. It has been proved that using images from moving cameras, it is feasible to estimate the camera poses $\{\hat{x}_{[k]}\}$ and 3D map points positions $\{\hat{X}_i\}$ by exploiting visual odometry (VO) approaches.

With additional information, the above visual SLAM problem can be simplified to other sub-problems in the same framework. If an autonomous vehicle navigate itself using detectable landmarks with known position, i.e., $\{\vec{X}_i\}$ is known, it is a landmark-based navigation problem which estimates the state vectors $\{x_{[k]}\}$ using the visual measurements. On the other hand, if the poses of the rover can be estimated reliably using other sensors, e.g., GNSS (Global Navigation Satellite System) receivers and IMUs (Inertial Measurement Units), the framework simplifies to a mapping problem.

As the camera moves with the vehicle, there will be consistently new image measurements. As a result, the Bayesian Network in Fig. 3.1 will grow over time, and meanwhile, the dimension of the optimization will significantly expand. To cope with the problem, the marginalization of the whole distribution is essential. A state-of-the-art solution is to use the extended Kalman filter (EKF) technique. The papers [38], [39] and [40] all describe EKF-based visual SLAM approaches. For the category of methods applying EKF, the pose of the camera at time k and the position of the 3D map points are stacked into a state vector. Through prediction and update phases using new measurements from time $k+1$, a marginalized optimal solution can be obtained. However, for the specific visual SLAM problem, EKF-based approaches have a few crucial problems. First of all, the EKF linearize the state-space and the measurement function at each step with the available knowledge. When better estimation is obtained as new measurements are fed in, there is no chance for relinearization for the past estimation. Secondly, the filter has to take all the map points into the state vector, so that the dimension of the state space increases significantly over long term, which makes the calculation of the matrix inversion too slow for practical usage. Last but not least, the true positions of static 3D feature points are actually constant and independent on each other. By stacking all the map points into the state vector, the positions become time-correlated during the filtering, which can generate biased estimation as the example in [41]. The consistency of EKF-based SLAM is analyzed in [42].

As an improvement, the particle filter (PF) technique uses Monto-Carlo method to estimate the probability density function, which does not require a linear state-space model. Therefore, by using a particle filter, the relinearization problem of EKF can be avoided. Montemerlo et al. propose a breakthrough algorithm FastSLAM in [43] that uses Rao-Blackwellized particle filter to solve the SLAM problem. Following the framework, Eade and Drummond proposed a monocular camera based SLAM algorithm in [44], which is shown to outperform the EKF based approaches in complexity to achieve similar accuracy. In [45], Sim et al. proposed a state of the art PF-based stereo SLAM method. However, although the Monto-Carlo based PF can cope with linearity in the measurement equations, the other problems of Bayesian filter based approaches still remain. For example, if the particle number is not sufficiently large (as a compromise for computational complexity), the particles will congregate after some resampling processes [46]. If the particles drift away from the true trajectory after particle congregation happens, the PF cannot be recovered from it. The reason behind is that the Bayesian filters are based on the hidden Markov model (HMM) as shown in Fig. 3.2. A HMM relies on two assumptions:

1) The state transition is a Markov process;

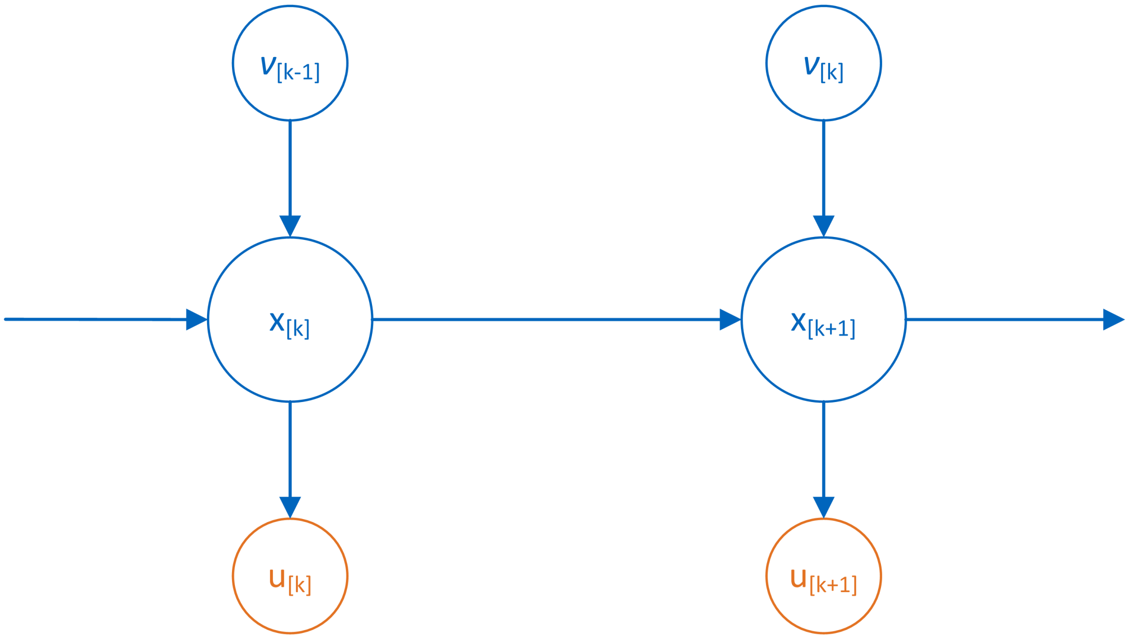


Figure 3.2: Hidden Markov model with control inputs

2) The measurements at time instant k are only dependent on the current state.

However, in the visual SLAM problem, the visual measurements are dependent both on the geometric states of the camera and on 3D feature points which are visible over multiple frames. The two assumptions of HMM are violated. Therefore, the Bayesian filters are not the optimal estimator for visual SLAM problems.

As an improved solution, the graph optimization based approaches are proposed. Instead of marginalizing out all the old states, graph-based methods optimize the whole trajectory with all the available information. However, the dimension of the parameter space is too high for real-time processing if all the frames are included. As a result, the graph based methods only use keyframes for the batch optimization. The criterion of the keyframe selection will be introduced in Section 3.2. Some recent Bayesian filter based VSLAM algorithm such as [47] is also keyframe-based, but the problems caused by the HMM still remain. To further reduce the computational complexity for the graph optimization, the approaches exploit the (approximate) sparsity characteristic of the graph information matrix to marginalize the inactive graph nodes, e.g., the map points that are invisible for a period of time. State of the art inference algorithms for the SLAM graph optimization are [48] and factor graph based [49]. Due to the advantages over Bayesian filter based approaches, we choose to use graph-based optimization in this work.

It should be mentioned that if only a single camera is available, the estimation of both the camera motion and the map point position has a global scale ambiguity, while a well calibrated stereo camera rig with sufficient baseline length does not have the problem. The state-of-the-art visual navigation methods based on monocular cameras and the scale problem are reviewed in detail in Section 4.1. In this Chapter we start with the methods using stereo camera rigs, which do not need to tackle with the scale problem yet.

3.2 Visual Navigation using a Stereo Camera Rig – a Review

Visual navigation using passive sensing vision is normally based on a calibrated stereo camera rig. With the aiding of the baseline information between the two cameras, the depth information lost in perspective projection can be reconstructed so that both the ego-motion trajectory and the 3D map can be estimated without any scale ambiguity. Because both the egomotion and the 3D location of the feature points are estimated, visual navigation provides a possible solution to the SLAM problem. In some applications which concern the pose estimation more than building a map, the onboard system will not save the map points given limited amount of storage. This leads to visual odometry approaches, e.g., the state-of-the-art approach proposed by Nister et al. in [50]. The visual odometry technique has been successfully applied in Mars exploration in the past years [1]. However, the visual odometry is based on dead-reckoning concept, so the pose estimation will drift from the true value over time, which is a crucial issue if one wants to move toward full autonomy. Using the stored map points, the camera system is able to detect the places it has visited before and add geometric constraints on the estimated parameters, which is normally referred as loop closing (from the concept of control loop). With loop closure, the accumulated estimation error can be significantly reduced. In this section, we will introduce the basic procedure of feature points based visual navigation methods. Related state of the art work in visual odometry and visual SLAM will also be reviewed. However, due to the huge amount of scholar work in the visual navigation field in the past years, it is rather challenging to provide a complete review including all important approaches. Therefore, we only mention those are most relevant to our work. As precious references, important approaches of visual odometry are reviewed in [51], and a thorough overview of the most recent visual SLAM work can be found in [52].

As defined in Section 2.3 and shown in Fig. 2.5, we choose to use well aligned stereo camera rigs with two identical cameras. The origin of the camera frame is defined at the projection center of the left camera, and both cameras have the same attitude. The position of the right camera is $\vec{b}^{(C)} = [b_x, 0, 0]^T$. $\Omega_L, \Omega_R \subset \mathbb{R}^2$ are the left and right virtual image planes, respectively. Applying the pinhole model, the perspective projection of point i to the left camera can be formulated as

$$\tilde{u}_{i,L} = d_i \begin{bmatrix} u_{i,L} \\ 1 \end{bmatrix}^T = K_C \vec{X}_i^{(C)}, \tag{3.14}$$

where $d_i = X_{i,z}^{(C)}$ is the depth of the point, and K is the camera intrinsic matrix. $u_{i,L} \in \mathbb{R}^2$ denotes the Cartesian coordinates of the point's two-dimensional (2D) location in the image, and $\tilde{u}_{i,L} \in \mathbb{P}^2$ is the corresponding homogeneous coordinates in the extended Euclidean space. The projection of the same point on the right camera is

$$\tilde{u}_{i,R} = d_i \begin{bmatrix} u_{i,R} \\ 1 \end{bmatrix}^T = K_C(\vec{X}_i^{(C)} - \vec{b}^{(C)}). \tag{3.15}$$

In Section 2.5, we defined navigation frame (N) as a fixed coordinate frame with its origin at the starting location of the rover. The projection of a point in the navigation frame is shown in Fig. 3.3. For a dynamic stereo rig with position $\vec{c}_{[k]}^{(N)}$ and attitude $R_{(k\to N)}$ in the navigation frame at time k, the 3D coordinates of a point in the camera frame (C) is related to that in the navigation frame (N) as

$$\vec{X}_i^{(C)} = \vec{X}_i^{(N)} - \vec{c}_{[k]}^{(N)}. \tag{3.16}$$

As a result, the feature location in the left and right views of the stereo rig at time instant k are respectively:

$$u_{i[k],L} = \pi(\vec{X}_i^{(N)}, R_{(k\to N)}, \vec{c}_{[k]}^{(N)}) = \frac{\begin{bmatrix} 1,0,0 \\ 0,1,0 \end{bmatrix} K_C R_{(N\to k)} \left(\vec{X}_i^{(N)} - \vec{c}_{[k]}^{(N)}\right)}{[0,0,1] K_C R_{(N\to k)} \left(\vec{X}_i^{(N)} - \vec{c}_{[k]}^{(N)}\right)} \tag{3.17}$$

$$u_{i[k],R} = \pi_R(\vec{X}_i^{(N)}, R_{(k\to N)}, \vec{c}_{[k]}^{(N)}) = \frac{\begin{bmatrix} 1,0,0 \\ 0,1,0 \end{bmatrix} K_C \left(R_{(N\to k)} \left(\vec{X}_i^{(N)} - \vec{c}_{[k]}^{(N)}\right) - \vec{b}^{(C)}\right)}{[0,0,1] K_C \left(R_{(N\to k)} \left(\vec{X}_i^{(N)} - \vec{c}_{[k]}^{(N)}\right) - \vec{b}^{(C)}\right)}. \tag{3.18}$$

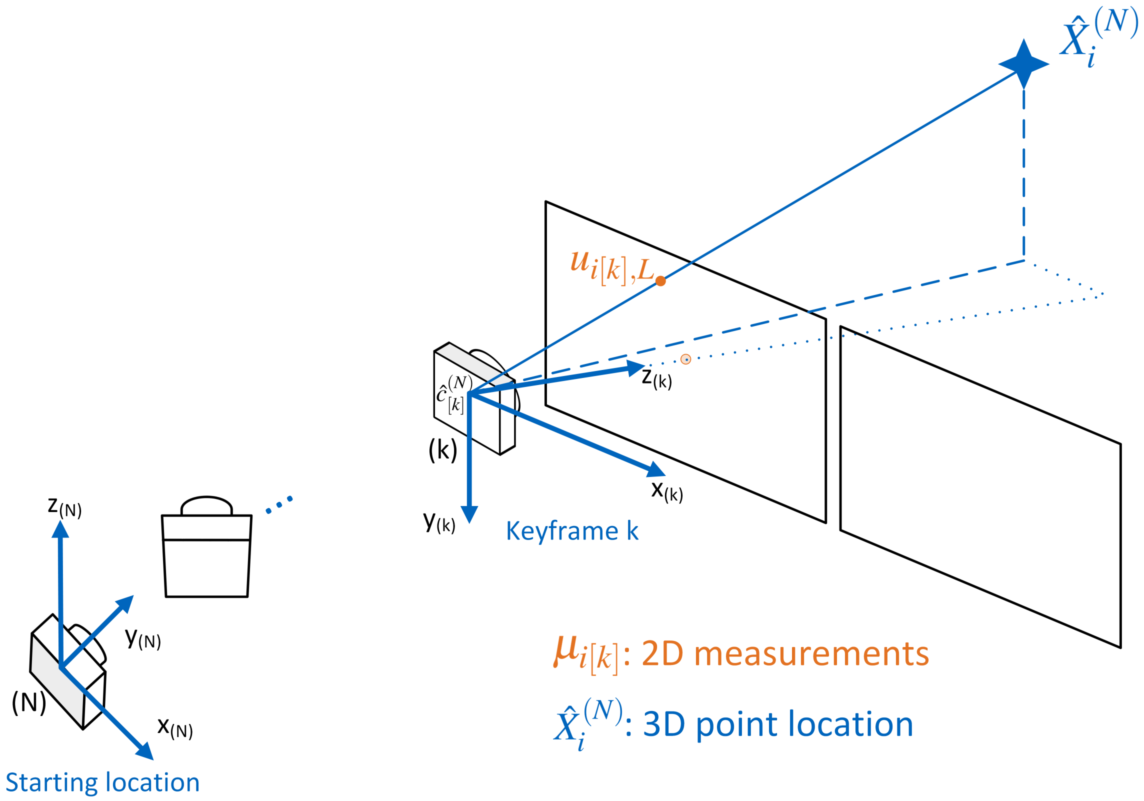


Figure 3.3: Projection of a point in navigation frame

As shown in Section 2.5, for a stereo rig mounted on a vehicle constrained to be moving in a plane, the pose can be parameterized by the state parameters $x_{[k]}^{(N)} = [c_{[k],x}^{(N)}, c_{[k],y}^{(N)}, \phi_{[k]}^{(N)}]^T$ as

$$\vec{c}_{[k]}^{(N)} = \begin{bmatrix} c_{[k],x}^{(N)} \\ c_{[k],y}^{(N)} \\ 0 \end{bmatrix}, R_{(N\to k)} = \begin{bmatrix} \cos(\phi_{[k]}^{(N)}) & -\sin(\phi_{[k]}^{(N)}) & 0 \\ 0 & 0 & -1 \\ \sin(\phi_{[k]}^{(N)}) & \cos(\phi_{[k]}^{(N)}) & 0 \end{bmatrix}. \tag{3.19}$$

The corresponding planar position of the rover is defined as $\vec{\beta}_k^{(N)} = [c_{[k],x}^{(N)}, c_{[k],y}^{(N)}]^T$.

We define a projection function for stereo rigs in planar motion: $\pi_S(\cdot) : (\mathbb{R}^3, \text{vec}(\mathfrak{se}(2))) \to \mathbb{R}^4$ for the point i and the vehicle pose at time k so that

$$u_{i[k]} = \begin{bmatrix} u_{i[k],L} \\ u_{i[k],R} \end{bmatrix} = \pi_S(\vec{X}_i^{(N)}, x_{[k]}^{(N)}) \in \mathbb{R}^4. \tag{3.20}$$

The corresponding noisy measurements are denoted by

$$\mu_{i[k]} = u_{i[k]} + n_{i[k]} \in \mathbb{R}^4, \tag{3.21}$$

with $E\{n_{i[k]}\} = \vec{0}$, $E\{n_{i[k]}n_{i[k]}^T\} = \Sigma_{u,i[k]}$. $E\{\cdot\}$ denotes the expected value function.

As reviewed in Section 3.1, there are direct and indirect visual navigation methods, and we focus on feature based indirect methods in our application. Feature points that are well distinguishable and trackable are crucial for visual odometry and VSLAM algorithms. Features are located in the images by feature detectors and are extracted into a feature descriptor vector. Normally the detectors and descriptors are designed jointly to achieve best performance. Widely used feature point detectors and descriptors includes the Harris corner detector [53], the Shi-Tomasi corner detector [54], the FAST feature detector [55], the BRIEF descriptor [56], the SIFT feature [37], the SURF feature [57], and the ORB feature [58]. Using feature detectors, several feature points can be matched between the stereo images and tracked over frames for a certain period of time.

Constrained by the stereo baseline and the projection equation Eqn. (3.17) and (3.18), the 3D position (in the camera frame) of any feature point matched between the left and right view can be estimated using triangulation methods. If the same feature is matched across two consecutive frames as the camera moves, the relative motion can be resolved by applying the geometric constraints [20] in rigid body motion. As a result, the motion tracking can be executed if a sufficient number of feature points can be matched and tracked. More specifically, to start the tracking, given a set of stereo measurements $\{\mu_{i,[1]} : i = 1, ..., N_p\}$ and the initial camera pose in the navigation frame $x_{[1]}^{(N)}$, the 3D position of the point i can be estimated by stereo triangulation as

$$\hat{X}_i^{(N)} = \pi_S^{-1}(\mu_{i[1]}, x_{[1]}^{(N)}). \tag{3.22}$$

Using $\{\mu_{i[2]} : i = 1, ..., N_p\}$, i.e., the feature location of the same point in the following frame, the relative motion between the two consecutive frames can be estimated. There are two ways of calculating the relative motion. The first category of approaches triangulate the 3D points location in the camera frame at frame number 2. Then the egomotion of the camera is extracted using the two matched groups of local point clouds. Typical work using such method is the visual odometry of Mars exploration rovers from NASA [1]. However, the error resides in both triangulated 3D locations degrades the accuracy of the motion estimation. Nistér et al. proposed a visual odometry method in [50] based on the minimization of 2D reprojection error from the 3D point to the second image plane, and indicated that it outperforms the point cloud matching methods. More generally, using N_p tracked features at frame k, the pose of the vehicle at time $k + 1$ can be estimated by minimizing the reprojection error

$$\hat{x}_{[k+1]}^{(N)} = \underset{x_{[k+1]}^{(N)}}{\arg\min} \sum_{i=1}^{N_p} \left\| \mu_{i[k+1]} - \pi_S(\hat{X}_i^{(N)}, x_{[k+1]}^{(N)}) \right\|_{\Sigma_{u,i[k+1]}^{-1}}^2, \tag{3.23}$$

where $\|\cdot\|_{\Sigma^{-1}}$ denotes the Mahalanobis distance with respect to covariance Σ. Using the estimated pose, the 3D position of the new features detected in frame $k + 1$ can be updated using $\pi_S^{-1}(\cdot)$. Consequently, the motion tracking can be continued as long as a sufficient number of features can be tracked in consecutive frames.

The problem that estimates the camera pose using n points with known 3D position and the corresponding 2D locations on the image plane is referred to as Perspective n-Point (PnP) problem. The PnP problem has been investigated by many authors. There are 6 DOF in the unknown camera pose, while each point provides two measurement equations. Hence the researches mainly focus on the cases that $n = 3, 4, 5$, since the cases $n < 3$ will result in an underdetermined system and the PnP problem with $n \geq 6$ points can be solved by simple direct linear transformation (DLT) algorithm [20]. In Nistér's visual odometry method in [50], 5-points (P5P) algorithm is applied. The 5-point algorithm from Nistér in [35] and that from Li and Hartley in [59] are the most widely used PnP methods in visual navigation. The 3-point algorithms, e.g., [60] and [61], have been proved to have non-unique (but finite) solutions. By adding a 4-th point, P4P methods such as [62] and [63] can obtain a unique pose solution, but there are more degenerated situations compared with 5-point algorithms. Another state of the art approach of general PnP problem is the EPnP method [64] from Lepetit et al., which is able to obtain an accurate solution with $O(n)$ computational complexity. In practice, PnP algorithms are normally used along with outlier rejection methods such as RANSAC [65], so that the mismatched features can be excluded from the calculation.

Since there is an error in the motion estimation at each frame, estimating the trajectory properly is essential to reduce the drift caused by undetected outliers and by error accumulation. Using Bayesian filters is an intuitive solution to the visual navigation problem. Due to the nonlinearity of the measurement model in Eqn. (3.17) and Eqn. (3.18), either an extended Kalman filter (EKF) or a particle filter can be used. The Bayesian filter optimizes the camera states estimation at time $k+1$ given the states at time k and all the measurements from beginning. A state of the art Bayesian filter based method is the visual odometry approach proposed by Kitt et al. in [66]. The method utilizes the geometry constraints among the four views in two consecutive stereo frames by using trifocal tensor and applies an EKF framework to track the camera motion consistently. Other important methods during the development of EKF based stereo visual navigation include [67], [68], and [69], etc. In [45], Sim et al. proposed a visual SLAM algorithm using Rao-Blackwellised particle filter, which outperforms EKF based approaches in the linearization error. In the Bayesian filter based methods, the latest camera pose and all the (active) map points are stacked into the state vector. As discussed in Section 3.1, the Bayesian filter approach essentially marginalizes the overall joint distribution by assuming a hidden Markov model so that all the old camera poses are discarded by exploiting the conditional independency. However, the assumption is not always valid. The linearization error in the measurement equation in the past cannot be corrected using the following measurements using Bayesian filters. Since the obtained motion estimates follow dead-reckoning principle, the estimation error due to unprecise linearization will accumulate over time.

The other category of methods propose to use a batch optimization to maximize the a posterior probability of the SLAM problem, which can be modeled by a pose graph. A state of the art approach is proposed by Dellaert and Kaess in [70]. The so called "Square Root SAM" algorithm exploits a factor graph model [71], [72] to deal with the probabilistic inference in the graph. Other important algorithms using graph based batch optimization include [73], [49], etc. In these methods, a global optimization for both 3D point position and the vehicle poses is performed using N_k keyframes and N_p map points:

$$\{\hat{x}^{(N)}_{[k]}, \hat{X}^{(N)}_i\} = \underset{\{x^{(N)}_{[k]}, \vec{X}^{(N)}_i\}}{\arg\min} \sum_{i=1}^{N_p} \sum_{k=1}^{N_k} S_{ik}(x^{(N)}_{[k]}, \vec{X}^{(N)}_i), \tag{3.24}$$

$$\text{with} \quad S_{ik}(x^{(N)}_{[k]}, \vec{X}^{(N)}_i) = v_{i[k]} \left\| \mu_{i[k]} - \pi_S(\vec{X}^{(N)}_i, x^{(N)}_{[k]}) \right\|^2_{\Sigma^{-1}_{u,ik}}, \tag{3.25}$$

where $v_{i[k]}$ is the binary visibility mask the same as defined in Section 3.1, which assumes $v_{i[k]} = 1$ if feature i is visible to the camera at time instant k, otherwise $v_{i[k]} = 0$. Therefore, by executing the optimization in Eqn. (3.24), the rover obtains a set of egomotion estimates expressed in its navigation frame, i.e., $\{\hat{x}^{(N)}_{[k]}\}$. Since the relation between the visual measurements and the pose parameters to be estimated is highly nonlinear, a coarse pose and map points' location estimation is required so that the batch optimization can converge to the correct global optimum. Therefore, the accuracy of the motion tracking in Eqn. (3.23) is crucial to the convergence of the graph optimization.

In order to accelerate the VSLAM, Klein and Murray proposed a breakthrough approach in [74] abbreviated as Parallel Tracking And Mapping (PTAM), which is designed for monocular camera based SLAM. In the PTAM method, the motion tracking and the global batch optimization are separated into two threads. The tracking thread can provide a fast coarse estimate of the parameters, while the mapping thread calculates accurate camera poses and a map with global consistency by exploiting batch optimization using only keyframes. As a result, the method enables a real time visual SLAM framework with sufficient robustness and accuracy. Many following state of the art approaches are based on the same structure, e.g., the widely used ORB-SLAM algorithm [75] and LSD-SLAM [34].

As the development of visual navigation, most recent state-of-the-art visual SLAM algorithms are keyframe-based. The difference between the images at two consecutive frames are normally quite small, provided that neither the movement of the rover is significantly fast nor the frame rate of the camera is significantly low. As a result, on the one hand, the information gain is limited for very similar consecutive images, so the computational power and the storage space can be saved by only processing keyframes. On the other hand, the motion estimation using images with tiny displacements has higher uncertainty, since the geometric noise and biases can be dominant in such cases. More detailed discussion can be found in Section 3.3. Keyframes are normally selected following two criterion:

1) The neighbor keyframes must have common field-of-view (FoV) and enough tracked common feature points;
2) The images of neighbor keyframes must be sufficiently different, i.e., there must be sufficient position displacement for tracked features.

By applying the keyframe selection, the visual odometry and visual SLAM methods can make the best effort to retain the drift low while keeping the processing power requirements in the feasible range.

An important difference between VO and VSLAM is the capability of detecting loop closure. By saving the parameters and the features at keyframes, VSLAM techniques are able to calculate the similarity between a new image and the old keyframes. If the similarity is significant, the robot can make the conclusion that the two images are representing the same scene, and the relative geometry can be estimated using the location of the common features. This process adds an a priori constraint between two camera pose nodes in the pose graph, so that the drift of the whole trajectory can be mitigated by closing the loop (following the feedback control loop philosophy). Most loop closure algorithms are based on bag of visual words (BoW) [76], which trains a visual vocabulary using feature descriptors and represents an arbitrary image using the histogram of the appeared visual words in the vocabulary. State of the art loop closure approaches include [77] and [78]. In [79], Williams et al. reviewed the state of the art methods of loop closing by then.

Loop closure is a powerful tool and can effectively mitigate the drifts accumulated in trajectory estimation. However, it requires to revisit mapped areas and to ensure a successful loop closure detection, the appearance of the environment should not change significantly after a while. These constraints have limited the application of the loop closing techniques in practice. A main contribution of this work is to have comparable performance as VSLAM without revisiting old places for loop closure by exploiting additional ranging measurements from radio links. The detailed sensor fusion methods are introduced in Chapter 4, 5 and 6.

3.3 General Uncertainty Model for Geometric Estimation using Vision Systems

In the field of navigation, it is crucial to model the uncertainty in pose estimation, so that the reliability of the estimation can be evaluated. Since the pose estimation and the mapping problem are tightly correlated, researchers proposed to use SLAM techniques to solve the joint problem. Therefore, the geometric estimation uncertainty of the camera poses and the map points position should be discussed jointly as well. In this section, we propose a generalized uncertainty model of the 3D reconstruction and pose estimation using 2D feature locations from multiple views. For the common sensor set-up of stereo camera rigs or monocular cameras, the uncertainty model can be easily specialized from the proposed general model. If the multiple views have particular geometric constraints, e.g., taken from a camera in planar motion, the corresponding uncertainty model is simply a specialized case of this general uncertainty model. As described in Section 2.2, for the projection from point $\vec{X}_i^{(W)}$ to camera at k-th view, the 2D location of a point in the image plane $\Omega_{[k]}$ is a function $\pi(\cdot)$ of the 3D point position $\vec{X}_i^{(W)}$ and camera extrinsic parameters, i.e., camera position $\vec{c}_{[k]}^{(W)} \in \mathbb{R}^3$ and attitude $R_{(k \to W)} \in \mathbf{SO}(3)$, which can be parameterized together as a 6 degrees of freedom (DOF) pose vector $x_{[k]} \in \mathbb{R}^6$. The relation can be denoted by

$$u_{i[k]} = \pi(\vec{X}_i^{(W)}, x_{[k]}).$$

Explicitly, the $x-$ and $y-$ components of the 2D feature point are expressed as:

$$u_{i[k],x} = \frac{[1,0,0]K_C R_{(W \to k)} \left(\vec{X}_i^{(W)} - \vec{c}_{[k]}^{(W)}\right)}{[0,0,1]K_C R_{(W \to k)} \left(\vec{X}_i^{(W)} - \vec{c}_{[k]}^{(W)}\right)} \tag{3.26}$$

$$u_{i[k],y} = \frac{[0,1,0]K_C R_{(W\to k)} \left(\vec{X}_i^{(W)} - \vec{c}_{[k]}^{(W)}\right)}{[0,0,1]K_C R_{(W\to k)} \left(\vec{X}_i^{(W)} - \vec{c}_{[k]}^{(W)}\right)} \tag{3.27}$$

with K_C as the intrinsic matrix of the camera.

Due to the presence of noise, in feature-based algorithms using the feature locations on the image plane as measurements, the accuracy of the estimated parameters is limited. The accuracy can be evaluated by the Cramér-Rao lower bound (CRLB) [80].

Assuming all the 2D measurements are independent and identically distributed (i.i.d.), by stacking all the parameters into a vector θ, the log-likelihood function of all the measurements given the parameters can be written as

$$\log\left(p(\mu|\theta)\right) = \sum_{q=1}^{N_Q} \log\left(p(\mu_q|\theta)\right),$$

with $\mu = \{\mu_{i[k]}|i = 1...N_p, k = 1...N_k\} = \{\mu_1, ..., \mu_q, ..., \mu_{N_Q}\}$ and state vector

$$\theta = \left[\vec{X}_1, ..., \vec{X}_{N_p}, x_1, ..., x_{N_k}\right]^T \in \mathbb{R}^{M\times 1}. \tag{3.28}$$

There are in total $M = 3N_p + 6N_k$ parameters in the vector θ ($M = 3N_p + 7N_k$ if the attitude is parameterized using 4 dimensional vectors such as unit quaternions). If the camera motion is constrained to be planar, the camera pose can be parameterized by a vector with 3 degrees of freedom. In such cases, the parameter vector has a reduced dimension of $M = 3N_p + 3N_k$. These dimensions change with different applications. A general uncertainty model regardless the dimensions is discussed here. According to the definition, the Fisher information matrix of estimating θ using μ can be written as

$$I_\theta = -E\left\{\nabla^2 \log\left(p(\mu|\theta)\right)\right\}.$$

The Hessian matrix of the measurements is calculated as following:

$$\nabla^2 \log\left(p(\mu|\theta)\right) = \begin{bmatrix} \frac{\partial^2 \log(p(\mu|\theta))}{\partial\theta_1^2} & \frac{\partial^2 \log(p(\mu|\theta))}{\partial\theta_1\partial\theta_2} & \cdots & \frac{\partial^2 \log(p(\mu|\theta))}{\partial\theta_1\partial\theta_M} \\ \frac{\partial^2 \log(p(\mu|\theta))}{\partial\theta_2\partial\theta_1} & \frac{\partial^2 \log(p(\mu|\theta))}{\partial\theta_2^2} & \cdots & \frac{\partial^2 \log(p(\mu|\theta))}{\partial\theta_2\partial\theta_M} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 \log(p(\mu|\theta))}{\partial\theta_M\partial\theta_1} & \cdots & \cdots & \frac{\partial^2 \log(p(\mu|\theta))}{\partial\theta_M^2} \end{bmatrix}.$$

As a result, using an unbiased maximum likelihood estimator, e.g., bundle adjustment, the parameter estimation accuracy is bounded by the following CRLB:

$$\text{var}(\theta) \geq \text{CRLB}(\theta) = I_\theta^{-1}.$$

The variance of each parameter is bounded by the diagonal terms of the inverse of the Fisher information matrix by

$$\text{var}(\theta_m) \geq (I_\theta^{-1})_{mm}.$$

It should be noted that since the projection function is non-linear, the calculation of the Hessian matrix is dependent on the value of the estimated parameters. Hence before exploiting the maximum likelihood estimator, an initial coarse estimate of the parameters are required in practice. In addition, the bounds of the attitude parameters are related to the parameterization used to represent the camera attitude, so the bound on attitude parameters would vary if a different parameterization was chosen.

In simplified scenarios that some of the parameters are known or can be estimated accurately using other sensors, the Fisher information matrix would be a submatrix of the above I_θ. For instance, if the positions of the tracked 3D landmarks $\{\vec{X}_i^{(W)}\}$ are known, the state vector only consists of the camera positions and attitudes, and the corresponding CRLB for the trajectory can be calculated. As another example, if the camera motion can be accurately estimated by a tracking system, the bound of the 3D reconstruction accuracy exploiting structure from motion (SFM) techniques can be obtained.

If we further assume (as in most state-of-the-art approaches) that the outliers in feature tracking are already removed using outlier rejection schemes such as RANSAC [65], so that the 2D feature location measurements of the inliers are multivariate Gaussian distributed variables. The likelihood function of the camera pose x_k and the location of the 3D feature point $\vec{X}_i^{(W)}$ given the 2D feature measurements is

$$p(\mu_{ik}|\vec{X}_i^{(W)}, x_k) = \frac{1}{2\pi}|\Sigma_u|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mu_{ik}-\pi(\vec{X}_i^{(W)},x_k))^T \Sigma_u^{-1}(\mu_{ik}-\pi(\vec{X}_i^{(W)},x_k))},$$

where

$$\Sigma_u = \begin{bmatrix} \sigma_x^2 & \varrho\sigma_x\sigma_y \\ \varrho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \tag{3.29}$$

is the covariance matrix of the 2D measurement. The corresponding log-likelihood function is

$$\log\left(p(\mu_{ik}|\vec{X}_i^{(W)}, x_k)\right) = \log\left(\frac{1}{2\pi}|\Sigma_u|^{-\frac{1}{2}}\right) - \frac{1}{2}(\mu_{ik} - \pi(\vec{X}_i^{(W)}, x_k))^T \Sigma_u^{-1} (\mu_{ik} - \pi(\vec{X}_i^{(W)}, x_k)).$$

The log-likelihood function of all the measurements given the parameters is

$$\log\left(p(\mu|\theta)\right) = -\frac{1}{2}\sum_{i=1}^{N_p}\sum_{k=1}^{N_k}(\mu_{ik} - \pi(\vec{X}_i^{(W)}, x_k))^T \Sigma_u^{-1} (\mu_{ik} - \pi(\vec{X}_i^{(W)}, x_k)).$$

The partial differences of the function with respect to the m-th parameter θ_m (under the Gaussian noise assumption) can be calculated by

$$\begin{aligned}\frac{\partial \log\left(p(\mu|\theta)\right)}{\partial\theta_m} = \sum_{i=1}^{N_p}\sum_{k=1}^{N_k}\Bigg(& W_{11}(\mu_{ik,x} - u_{ik,x})\frac{\partial u_{ik,x}}{\partial\theta_m} + W_{12}(\mu_{ik,y} - u_{ik,y})\frac{\partial u_{ik,x}}{\partial\theta_m} \\ & + W_{21}(\mu_{ik,x} - u_{ik,x})\frac{\partial u_{ik,y}}{\partial\theta_m} + W_{22}(\mu_{ik,y} - u_{ik,y})\frac{\partial u_{ik,y}}{\partial\theta_m}\Bigg)\end{aligned}$$

with the weighting matrix $W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} = \Sigma_u^{-1}$.

As a result, for Gaussian noise, the entries of the Fisher information matrix can be calculated analytically by

$$\begin{aligned}[I_\theta]_{m_1 m_2} &= -E\left\{\frac{\partial^2 \log\left(p(\mu|\theta)\right)}{\partial\theta_{m_1}\partial\theta_{m_2}}\right\} \\ &= \sum_{i=1}^{N_p}\sum_{k=1}^{N_k}\left(W_{11}\frac{\partial u_{ik,x}}{\partial\theta_{m_1}}\frac{\partial u_{ik,x}}{\partial\theta_{m_2}} + W_{12}\frac{\partial u_{ik,x}}{\partial\theta_{m_1}}\frac{\partial u_{ik,y}}{\partial\theta_{m_2}} + W_{21}\frac{\partial u_{ik,y}}{\partial\theta_{m_1}}\frac{\partial u_{ik,x}}{\partial\theta_{m_2}} + W_{22}\frac{\partial u_{ik,y}}{\partial\theta_{m_1}}\frac{\partial u_{ik,y}}{\partial\theta_{m_2}}\right) \\ &= \sum_{i=1}^{N_p}\sum_{k=1}^{N_k}\left(\frac{\partial\pi(\vec{X}_i^{(W)}, x_k)}{\partial\theta_{m_1}}\right)^T \Sigma_u^{-1}\left(\frac{\partial\pi(\vec{X}_i^{(W)}, x_k)}{\partial\theta_{m_2}}\right).\end{aligned} \tag{3.30}$$

If the FIM is a full-rank diagonal matrix, i.e., $[I_\theta]_{m_1 m_2} = 0 \iff m_1 \neq m_2$, the variance of the m-th estimated parameter is bounded by

$$\text{var}(\theta_m) \geq \left[\sum_{i=1}^{N_p}\sum_{k=1}^{N_k}\left(\frac{\partial\pi(\vec{X}_i^{(W)}, x_k)}{\partial\theta_m}\right)^T \Sigma_u^{-1}\left(\frac{\partial\pi(\vec{X}_i^{(W)}, x_k)}{\partial\theta_m}\right)\right]^{-1}, \tag{3.31}$$

which is equivalent to the classic uncertainty linear propagation for Gaussian variables. In most state-of-the-art uncertainty analysis work such as [81], [82] and [83], the spatial uncertainty is analyzed by linearizing the function between the measurement space and the parameter space, followed by a Gaussian distribution covariance propagation. However, it can be easily concluded from Eqn. (3.26) and Eqn. (3.27) that the FIM of the general visual SLAM problem is only block-diagonal but non-diagonal. Therefore, under the Gaussian noise assumption, it is more precise to analyze the spatial uncertainty using the CRLB than traditional covariance propagation methods.

3.4 Performance Limitation of a Short-baseline Stereo Rig

For a stereo camera rig, 2D feature points in the left and right images, which are projected from the same 3D point in the space, can be paired using feature matching methods. Then, the depth of the feature point can be reconstructed using the 2D locations measurements. When estimating the depth of a pair of matched 3D point using Eqn. (2.29), it is intuitive that the noisy 2D measurements will result in spatial uncertainty in 3D reconstruction.

Let us assume a stereo rig with the same and well aligned left and right cameras. The baseline length is b_x. The noisy 2D geometric measurements of a pair of matched feature points can be denoted as $\mu_{i,L} = u_{i,L} + n_{uiL}$ and $\mu_{i,R} = u_{i,R} + n_{uiR}$. The measurement noise distributions are assumed to be zero mean Gaussian with noise covariance

$$E\{n_{uiL}n_{uiL}^T\} = E\{n_{uiR}n_{uiR}^T\} = \Sigma_{ui} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}. \tag{3.32}$$

In such condition, the measurement disparity $\delta\mu_x = \mu_{ix,L} - \mu_{ix,R} = \delta x + n_{\delta x}$ is also Gaussian distributed with mean value $\delta x = u_{ix,L} - u_{ix,R}$ and covariance $2\sigma_x^2$. The depth value is inverse to the Gaussian random variable as

$$\hat{d} = \frac{f}{\delta\mu} b_x. \tag{3.33}$$

It has been proved in [84] that the mean and covariance value do not exist for the inverse of a Gaussian variable. Hence in order to analyze the spatial uncertainty of the depth estimator, we linearize the estimation function by the first order approximation of the Taylor expansion as

$$\hat{d} = \frac{fb_x}{\delta x} - \frac{fb_x}{\delta x^2}(\delta\mu - \delta x) = d + J_{\delta\mu}(\delta\mu - \delta x), \tag{3.34}$$

where $J_{\delta\mu} = -\frac{fb_x}{\delta x^2}$ is the gradient of the depth with respect to the disparity value. As a result, under the first order approximation, the estimated depth is also a Gaussian distributed random variable with mean value as the true depth and variance

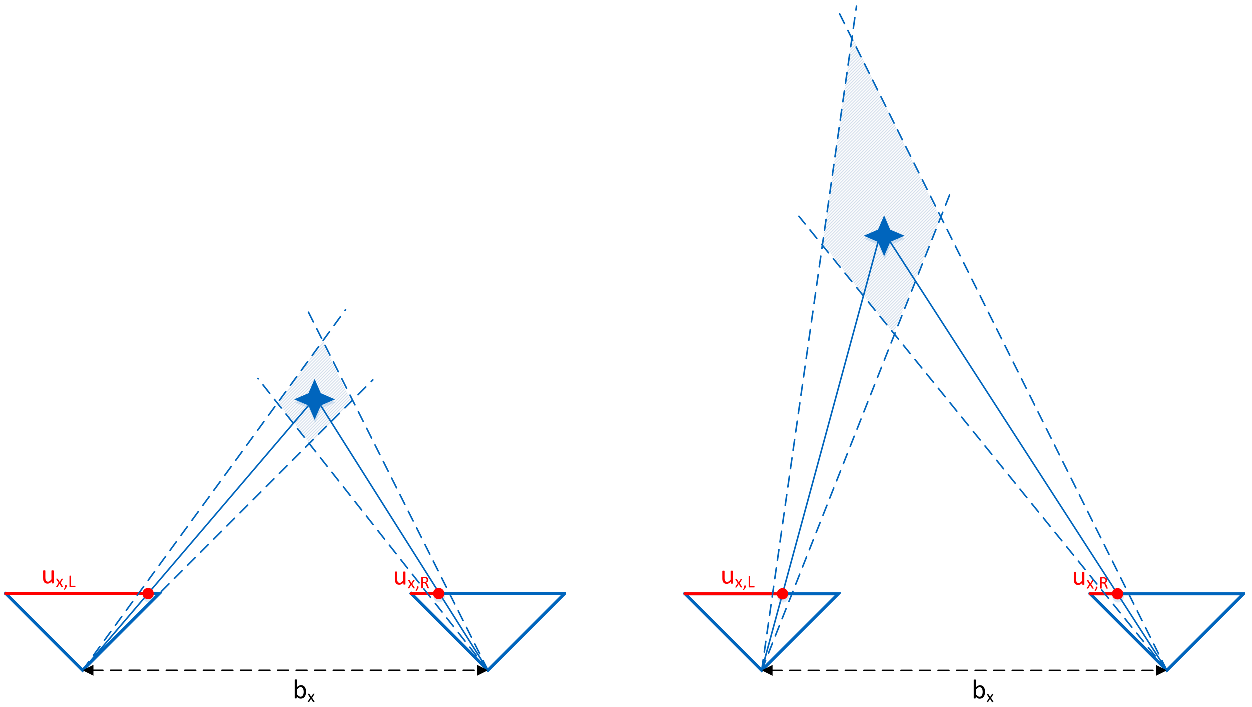
$$E\{(\hat{d} - d)^2\} = (J_{\delta\mu})^2 E\{(\delta\mu - \delta x)^2\} = 2\left(\frac{fb_x}{\delta x^2}\right)^2 \sigma_x^2. \tag{3.35}$$

According to Eqn. (3.31) in Section 3.3, such approximated variance is equivalent to the CRLB of the estimated parameter in one dimensional case. The corresponding standard deviation of the estimated depth is

$$\sigma_d = \frac{\sqrt{2}fb_x}{\delta x^2}\sigma_x = \frac{\sqrt{2}d}{\delta x}\sigma_x = \frac{\sqrt{2}d^2}{fb_x}\sigma_x. \tag{3.36}$$

The following term in Eqn. (3.36) is the dilution of precision (DOP) for the depth estimation, which reflects the geometric impact on the estimate:

$$dDOP = \frac{\sqrt{2}d^2}{fb_x}. \tag{3.37}$$



(a) Near feature point

(b) Far feature point

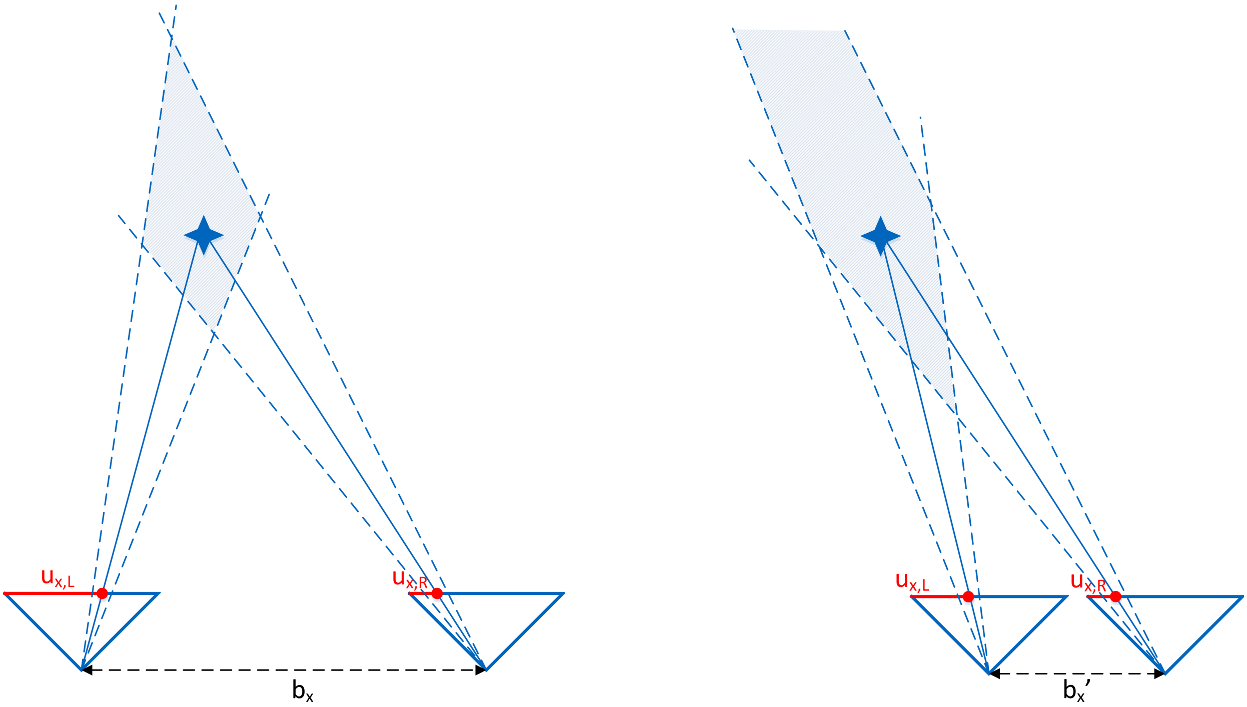
Figure 3.4: Examples of depth spatial uncertainty

It can be seen that for a specific stereo camera rig, if the optical lens is fixed, the 2D feature location uncertainty is propagated to the depth estimation uncertainty with a factor quadratic to the true depth, and inversely proportional to the baseline length. Therefore, for a stereo rig with fixed baseline length, the depth estimation of a close feature point has lower uncertainty compared with a further one. On the other hand, for a feature point with specific depth in the space, a short baseline stereo rig has larger spatial uncertainty in depth estimation and 3D reconstruction.

Fig. 3.4 provides a qualitative example of the impact of the feature point depth on the spatial uncertainty of the stereo reconstruction. In both subplots, the $x - z$ plane of the camera coordinate frame is shown, and the baseline length as well as the 2D feature position covariance are the same. In the Fig. 3.4a, the feature point is close to the stereo camera rig. The spatial uncertainty of the estimated 3D position is bounded in a relatively small area. At the same time, for a feature point which is distant from the stereo rig, as shown in Fig. 3.4b, the spatial uncertainty, especially the depth error is much larger.

An example of the baseline length impact on the stereo reconstruction uncertainty is given in Fig. 3.5. For a feature point in specific distance, a short baseline stereo rig may have huge spatial uncertainty in depth estimation and 3D reconstruction.

Take the Bumblebee2 [85] stereo camera's parameters as a realistic example. The well aligned stereo camera rig consists two identical cameras with $1/3$ inch imaging sensor and 1024×768 resolution. The baseline length is 0.12 [m], and the optical lens has 2.5 [mm] focal length which is equivalent to 550 [pixel] in the image plane. According to Eqn. (2.32), the theoretical minimal stereo sensing distance of the rig is 0.1289 [m]. If the 2D feature location measurements have noise with 1 [pixel] standard deviation in the x-axis, the relation between the feature point distance to the image plane d and the stereo depth estimation uncertainty σ_d is plotted in Fig. 3.6. It can be observed from the curve that for a point which is 3 meters away from the image plane, the depth reconstruction uncertainty using a Bumblebee2 stereo rig is around 20 centimeters. When the distance increases to 4 meters, the depth has a spatial uncertainty of 35 centimeters, which is already comparable to the distance between two consecutive keyframes in state-of-the-art VSLAM



(a) Long baseline

(b) Short baseline

Figure 3.5: Baseline length impact on spatial uncertainty for specific depth

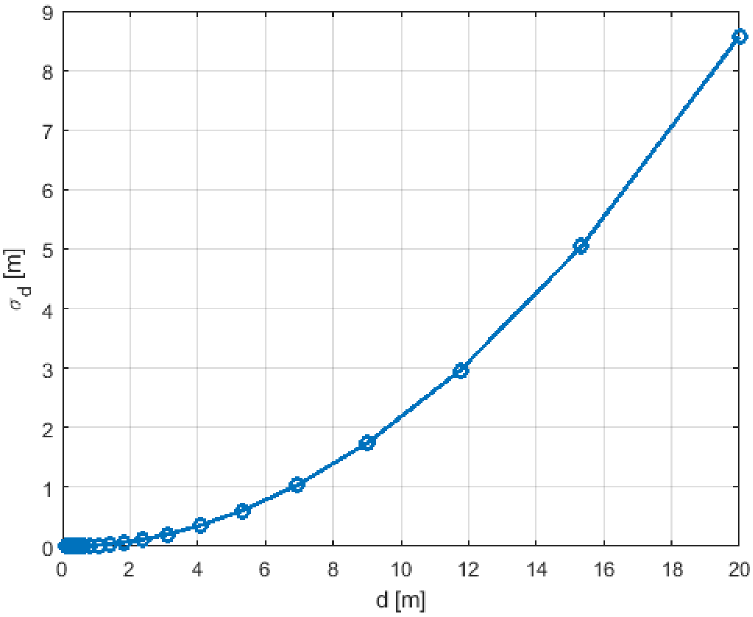


Figure 3.6: Uncertainty of the depth estimation with respect to the distance. $\sigma_x = 1$ [pixel], $b_x = 0.12$ [m].

algorithms. For feature points that are 7 meters away, the uncertainty already rises to 1 meter. The 3D reconstruction becomes too unreliable for the stereo rig with only 12 centimeters baseline length.

According to Eqn. (2.30), in stereo rig based 3D reconstruction, the geometric error in the $z-$axis of the camera frame propagates to the other two dimensions. Therefore, the uncertainty of the depth estimation is the crucial problem in stereo vision. If the covariance of the depth is high, the 3D reconstruction of the map points becomes unreliable for navigation applications.

In conclusion, reconstruction uncertainty for certain depth increases reversely with respect to the baseline of the stereo rig. If the baseline is short, for the far-away objects and feature points, the depth estimation will be highly unreliable. Furthermore, due to the resolution constraints of the camera sensor, the quantization error in sensing will be so large that it has no more advantage over a monocular camera. As a result, the weight, costs and computational complexity can be saved by using a monocular camera while without losing too much performance, in the situation that the features are not close but the possible stereo rig baseline length is limited. Hence in outdoor scenarios such as Mars exploration, it is more favorable to equip a monocular camera instead of a stereo rig for a robotic swarm agent with small size. Due to the above reasons, we choose to discuss the VSLAM problem for a swarm of cooperative small rovers equipped with monocular cameras, and propose our contributions based on such sensor set-up.

Moreover, two independently moving monocular cameras are more flexible than a stereo camera rig with a fixed baseline. In an exploration mission, the visible scenarios are always changing. For a stereo system with certain field of view, if a feature point locates too close with respect to the baseline length, it will not be observed by both cameras, or suffers from strong distortion at the image border areas even if it locates in the common field of view. On the other hand, if a feature point is distant from the stereo system, the stereo system will have few advantage over a monocular camera, as we discussed earlier. As a result, a fixed baseline cannot guarantee good performance in various scenarios. As an improvement, using two monocular cameras equipped on two independent mobile vehicles has more flexibility. Given a reliable relative pose estimation, which will be discussed in Chapter 5, the two cameras can form a flexible stereo vision system. If some landmarks or important feature points are far away, the two swarm elements can increase the baseline length of the stereo system to improve the estimation accuracy. If the landmarks are very close, the vehicles can move and adjust poses to reduce the baseline length and improve the co-observability. The common field of view detection method will be introduced in Section 5.3.

4. Single Vehicle Navigation using a Monocular Camera and a Ranging Radio Link

It has been introduced in the previous chapter that cameras play a significant role in autonomous navigation of robotic platforms. VSLAM methods based on stereo camera rigs can estimate the poses of the platform and map point locations with respect to the starting position. However, due to constraints on size, weight, accommodation and cost, a stereo rig can often not be implemented on small rovers in robotic swarms. Besides, the spatial uncertainty of stereo rig reconstruction has been discussed in detail in Section 3.4, and it can be concluded that if the baseline length of the stereo rig is much smaller than the distance from the image plane to the feature point, a stereo rig will not have significant advantage over a monocular camera. As a result, in our application of using robotic swarms to explore Mars, we focus on developing methods based on monocular cameras instead of stereo rigs.

However, due to the depth information is lost in the perspective projection for a single camera, the scale information of the camera motion as well as the environment is unknown. The scale problem consists of two challenges. First, the global scale cannot be reconstructed without other sensors or prior knowledge. Secondly, though the relative scales between two consecutive motion steps can be estimated, the error in the estimation propagates exponentially in the multiplication of the relative scales. Consequently, as time passes, the estimated scale with respect to that at the beginning can be significantly wrong. This problem is known as scale drift. In this chapter, the scale problem of monocular vision is reviewed in detail. Then, we propose a method to solve both challenging scale problems using range measurements from a single radio link between the monocular camera and an anchor point, e.g., a swarm element in static mode, or a base station.

4.1 Visual Navigation with Scale Ambiguity using a Monocular Camera – a Review

The monocular camera based pose estimation problem was first investigated by the computer vision community, who are interested in 3D reconstruction of a scene using a set of images taken from multiple views. Instead of using multiple cameras with accurately known extrinsic parameters, i.e., relative poses, one can use a single monocular camera moving around the object to be reconstructed, and estimate the camera motion as well as the 3D points location by utilizing the images taken from different view points. This is the structure from motion (SFM) concept. There are a few state-of-the-art SFM methods such as [86], [87] and [88]. The SFM approaches match common features from two image views, and estimate relative pose as well as the 3D location of the matched feature points according to epipolar geometry. Following the procedure, new images are added one after another. At last, a global least batch optimization is executed to refine the estimation results. There are several SFM methods that allow to use uncalibrated cameras, for example, the work in [89], [90] and [91]. However, in our applications, we can calibrate the camera before the mission, so we always assume that the camera intrinsic matrix K_C is known and remains the same for all the views. For a calibrated camera, the problem behind SFM methods is mathematically the same as the monocular visual SLAM problem, except for higher real-time processing requirements.

In the rest part of this section, we briefly introduce the epipolar geometry and the corresponding motion estimation method for VSLAM applications. Then, the scale problem is discussed, and we clarify the definition of global scale and relative scale in this thesis. At last, a few related state-of-the-art monocular VSLAM literatures are reviewed and discussed.

Unlike stereo systems with well-calibrated baseline information, the relative pose between any two views taken from a monocular camera must be estimated from visual cues. Given two images with common field of view of the same scene, the common points of interest visible in both views can be matched by

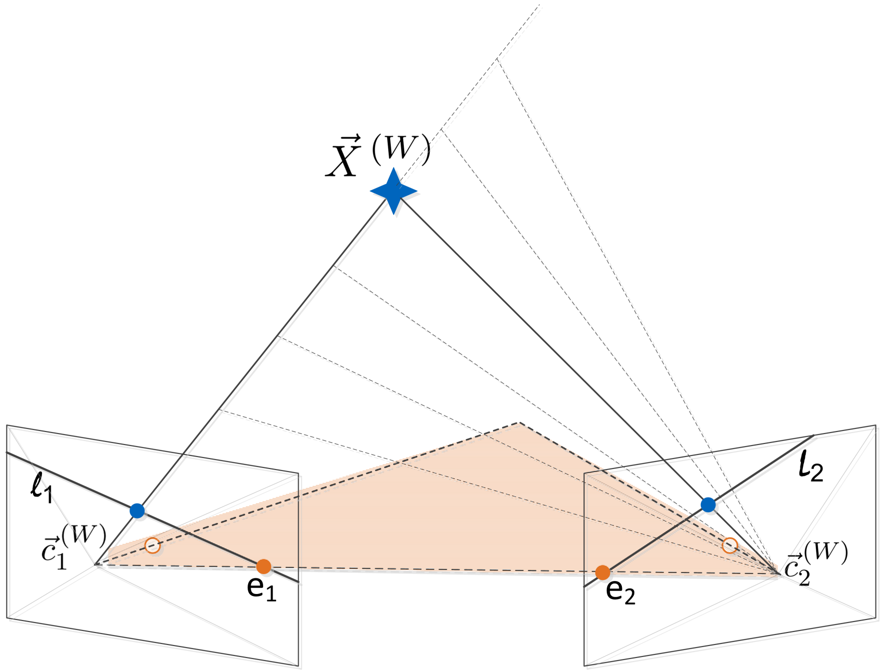


Figure 4.1: Epipolar geometry of two views

feature detection and feature matching. The location of the feature points in the image planes follows epipolar constraints, which is discussed in detail in [20]. As a basic introduction of it in our application, Fig. 4.1 illustrates the epipolar constraint in two view geometry. As introduced in Section 2.2, for a camera located at $\vec{c}_1^{(W)} \in \mathbb{R}^3$ with attitude $R_{(C_1\to W)}$ in the reference frame (W), an arbitrary 3D point in the space $\vec{X}^{(W)} \in \mathbb{R}^3$ is projected to the camera image plane as

$$\tilde{u}_1 = P_1\tilde{X}^{(W)} = K_C R^T_{(C_1\to W)}(\vec{X}^{(W)} - \vec{c}_1^{(W)}), \tag{4.1}$$

where $\tilde{u}_1 \in \mathbb{P}^2$ denotes the 2D feature location in homogeneous coordinates, $\tilde{X}^{(W)}$ the homogeneous coordinates of $\vec{X}^{(W)}$, and $P_1 \in \mathbb{R}^{3\times 4}$ is the projection matrix. As a reverse problem, the 2D feature point can be back-projected to the 3D space as a line due to the unawareness of the depth. The 3D coordinates of the candidates location can be expressed as

$$\tilde{X}^{(W)} = P_1^+\tilde{u}_1 + \tilde{c}_1^{(W)} \in \mathbb{P}^3, \tag{4.2}$$

where $(\cdot)^+$ denotes the pseudoinverse of the matrix. The camera location in homogeneous coordinates $\tilde{c}_1^{(W)} \in \mathbb{P}^3$ lies in the null space of the projection matrix P_1, so the projected 2D location is invariant to the change of the scaler factor ς. When the monocular camera moves (or more generally, for another camera view with projection matrix P_2 and projection center $\vec{c}_2^{(W)} \in \mathbb{R}^3$), the 3D point candidate positions can be re-projected to the second image plane as a line l_2. This line is called epipolar line. The location of the same feature point on the second view must lie on the epipolar line, which provides a correspondence relation between the location of a same feature point from two different views. It should be mentioned that e_2, the projection of the first camera center, locates on all the epipolar lines in the image plane, which is called epipole. The equation of the epipolar line l_2 can be expressed by the cross product of the epipole coordinates $\tilde{e}_2 \in \mathbb{P}^2$ and the reprojected feature location for $\varsigma = 0$ as $P_2P_1^+\tilde{u}_1$. The coordinates in the second image plane from the same feature point must fulfill the relation

$$\tilde{u}_2^T l_2 = \tilde{u}_2^T[e_2]_\times P_2P_1^+\tilde{u}_1 = \tilde{u}_2^T F\tilde{u}_1 = 0, \tag{4.3}$$

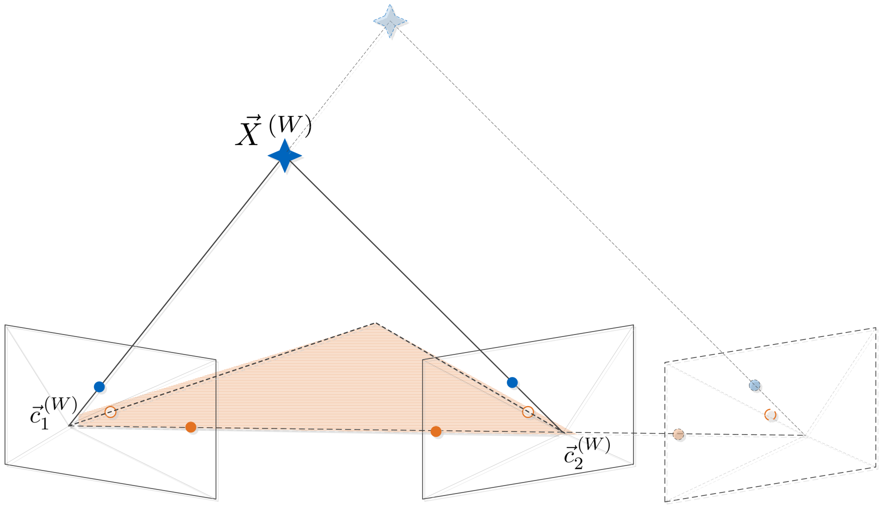


Figure 4.2: Scale ambiguity in the estimated translation

where $[\cdot]_\times$ is the skew symmetric constructor introduced in Appendix, and $F = [e_2]_\times P_2 P_1^+$ is the well known fundamental matrix which constrains the 2D locations of the same feature point in two image planes. Since the intrinsic parameters of both views are known and are the same, the two view geometry can be described by the epipolar constraint for calibrated cameras as

$$(K_C^{-1}\tilde{u}_2)^T E(K_C^{-1}\tilde{u}_1) = 0, \tag{4.4}$$

where $E = K_C^T F K_C$ is called essential matrix, which is only determined by the relative pose of the two views. The essential matrix has one zero eigenvalue, and two equal non-zero eigenvalues. In addition, the constraints are not violated if the matrix E is scaled by an arbitrary scalar factor, which is also reflected by the epipolar constraint from Eqn. (4.4). There are only 5 degrees of freedom in E. (See [20] for more details.) As a result, the essential matrix can be estimated given $N_p \geq 5$ pairs of matched features. Without loss of generality, if we take the camera frame of the first view (C_1) as reference frame, the two projection matrix can be represented in such case as

$$P_1 = K_C[I_3|\vec{0}], \quad P_2 = K_C[R_{(C_1 \to C_2)}|\vec{t}_{(C_1 \to C_2)}]. \tag{4.5}$$

Consequently, E can be decomposed as

$$E = [\vec{t}_{(C_1 \to C_2)}]_\times R_{(C_1 \to C_2)}, \tag{4.6}$$

where (C_1) and (C_2) are the camera frame for the two views respectively. As a result, the relative pose of the two views can be extracted from the subspace of the matrix using singular value decomposition (SVD). However, the extracted translation $\vec{t}_{(C_1 \to C_2)}$ is ambiguous by a positive scale factor, which is illustrated in Fig. 4.2. It can be seen that the measurements in the two camera views are invariant to the scale change of the whole scene. Given the 2D location of the feature point in two camera views, the metric length of the translation baseline is proportional to the depth of the point in 3D space.

The original concept of scale in computer vision is used to describe the metric size of a scene. As shown in Fig. 4.3, two features points are detected in the image as the vertices of an object. The real size of the object is dependent on the distance between the camera and it. The image can represent either a small item close by, or a large item far away. The scale of the visible scene from an image is proportional to the depth of the object points. Consequently, the previous stated ambiguity of baseline length $\|\vec{t}_{(C_1 \to C_2)}\|$, which is also proportional to the depth of the matched feature points, is referred as the scale ambiguity in monocular VSLAM.

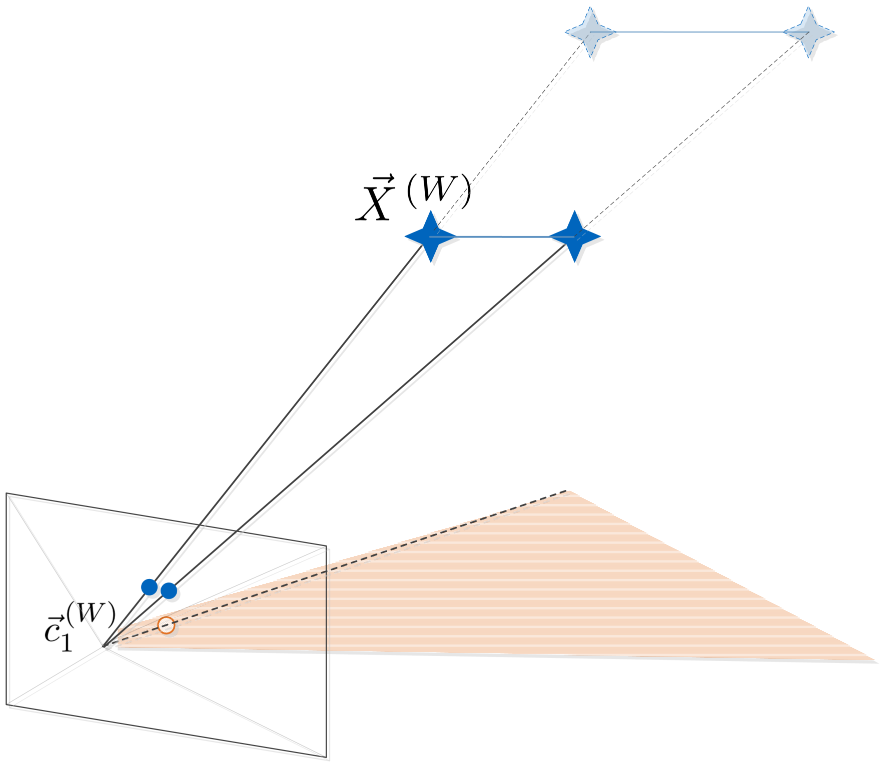


Figure 4.3: Feature depth and scale of a scene

Without a priori information about the scale of the observed scene, which is the ordinary situation in exploration missions, one cannot estimate the absolute length of the translation between two camera views

$$s_{(1\to 2)} = \|\vec{t}_{(C_1\to C_2)}\| = \|\vec{c}_2^{(W)} - \vec{c}_1^{(W)}\|. \tag{4.7}$$

Nevertheless, by discomposing essential matrix E in Eqn. (4.6) using SVD with multiple matched feature points, the direction of the translation can be calculated as a unit basis vector

$$\vec{\epsilon}_{(C_1\to C_2)} = \frac{\vec{t}_{(C_1\to C_2)}}{\|\vec{t}_{(C_1\to C_2)}\|}. \tag{4.8}$$

Applying $\vec{\epsilon}_{(C_1\to C_2)}$ as the baseline between two views, the 3D location of a matched feature point can be triangulated as

$$\hat{X} = \pi^{-1}(\mu_1, \mu_2, \vec{\epsilon}_{(C_1\to C_2)}, R_{(C_1\to C_2)}), \tag{4.9}$$

where μ_1, μ_2 are noisy measurements of the 2D feature location $u_1, u_2 \in \mathbb{R}^2$.

If the images are taken consecutively from the moving camera, the camera pose of the new image can be estimated with respect to the previous frames. Following the two-view epipolar geometry, the baseline between the 2nd and the 3rd image view is estimated as

$$\vec{\epsilon}_{(C_2\to C_3)} = \frac{\vec{t}_{(C_2\to C_3)}}{\|\vec{t}_{(C_2\to C_3)}\|}. \tag{4.10}$$

However, if the same feature point can be tracked over frames, additional constraints on the relative length between the translations is introduced. Fig. 4.4 illustrates such constraints with a simple example. Since the depth of the point with respect to image 2 is the same in the geometry of image pair 1-2 and image pair 2-3, the relative translation length

$$s_{r,2} = \frac{s_{(2\to 3)}}{s_{(1\to 2)}} \tag{4.11}$$

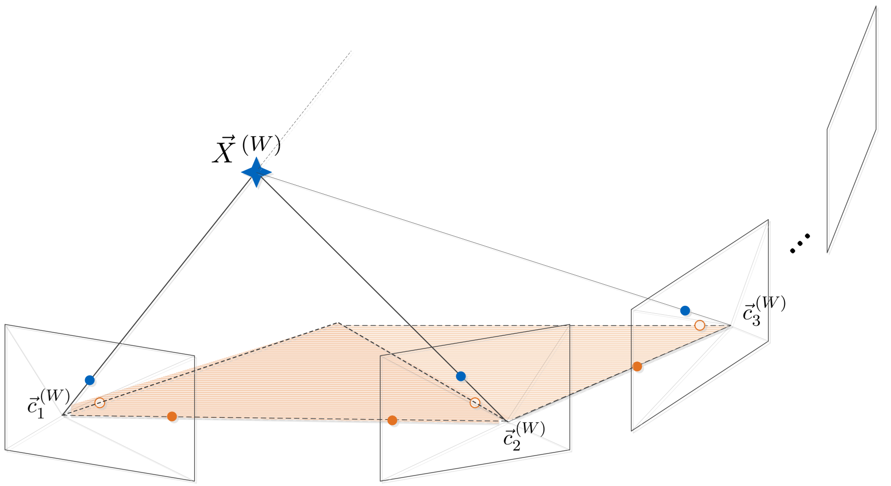


Figure 4.4: Relative scale of consecutive frames

is fixed, even if the absolute length is unknown. As a result, the whole set that contains the estimated camera poses and the corresponding triangulated 3D feature point positions only has a global scale ambiguity s_g. Without loss of generality, we can assume the first estimated translation has unit length and map it into s_g, so that the global scale can be defined as

$$s_g = s_{(1\to 2)} = \|\vec{c}_2^{(W)} - \vec{c}_1^{(W)}\|. \tag{4.12}$$

In Section 2.5, we defined navigation frame (N) as a fixed coordinate frame with its origin at the starting location of the rover. For monocular case, since the global scale is unavailable, we assume that the estimated motion satisfies

$$l_{(1\to 2)} = \|\vec{c}_2^{(N)} - \vec{c}_1^{(N)}\| \overset{!}{=} 1, \tag{4.13}$$

so that all the camera and map point positions in the navigation frame (N) are all scaled from the metric value in world frame by s_g. For a moving camera, the length of translation at time k can be expressed as

$$s_{(k\to k+1)} = \|\vec{c}_{[k+1]}^{(W)} - \vec{c}_{[k]}^{(W)}\| = s_g\|\vec{c}_{[k+1]}^{(N)} - \vec{c}_{[k]}^{(N)}\| = s_g \prod_{\kappa=2}^{k} s_{r,\kappa} l_{(1\to 2)} = s_g \prod_{\kappa=2}^{k} s_{r,\kappa}. \tag{4.14}$$

Consequently, we can express the monocular camera based motion tracking and feature triangulation in navigation frame as

$$\hat{x}_{[k+1]}^{(N)} = \underset{x_{[k+1]}^{(N)}}{\arg\min} \sum_{i=1}^{N_p} \left\| \mu_{i[k+1]} - \pi(\hat{X}_i^{(N)}, x_{[k+1]}^{(N)}) \right\|^2_{\Sigma^{-1}_{u,i[k+1]}}, \tag{4.15}$$

where $x_{[k+1]}^{(N)} \in \mathbb{R}^6$ consists both the camera position $\vec{c}_{[k+1]}^{(N)}$ and the rotation parameters at time k. N_p is the total number of the tracked feature points, and $\mu_{i[k+1]}$ denotes the 2D feature location measurements with covariance $\Sigma_{u,i[k+1]}$ for feature point i.

Following the basic concept, several monocular VSLAM approaches have been developed in the past years. Motion tracking using Bayesian filters is a straightforward and effective solution. Davison et al. proposed a monocular SLAM framework based on EKF (Extended Kalman Filter), for jointly estimating the camera states and the map [38]. In [92], Civera, Davison and Montiel propose to parameter the 3D point using inverse depth. The prameterization fits the Gaussian noise assumption of EKF better than conventional methods. Other important Bayesian filter based work include [93], [94], [95] and [47].

Another significant cluster of approaches executes the motion tracking in a simple dead reckoning way, and uses multiple view optimization across various frames to estimate the camera poses and the position of the map points more accurately. The optimization is normally referred as "bundle adjustment" in literatures, which minimizes the overall reprojection error by:

$$\{\hat{x}_{[k]}^{(N)}, \hat{X}_i^{(N)}\} = \underset{\{x_{[k]}^{(N)}, \vec{X}_i^{(N)}\}}{\arg\min} \sum_{i=1}^{N_p} \sum_{k=1}^{N_k} v_{i[k]} \left\| \mu_{i[k]} - \pi(\vec{X}_i^{(N)}, x_{[k]}^{(N)}) \right\|_{\Sigma_{u,i[k]}^{-1}}^2 , \tag{4.16}$$

where $v_{i[k]}$ is the visibility indicator for the i-th point in the k-th view. A breakthrough work using bundle adjustment is the Parallel Tracking and Mapping (PTAM) algorithm [74] developed by Klein and Murray, which divides the tracking and mapping into separate threads to accelerate the computation. On the one hand, if all the frames are included in the optimization, the computational complexity is too high for realtime processing. On the other hand, two close frames do not provide significantly different information. As a result, PTAM uses only keyframes for global optimization, while the other frames are only used for motion tracking. The algorithm shows good performance in both accuracy and speed. A large number of following approaches are based on the PTAM framework, e.g., the work [96] and [97]. The state-of-the-art ORB-SLAM method from Mur-Artal, Montiel and Tardós [98] is also developed based on the PTAM framework. The method utilizes ORB features [58] and a few novel techniques. It is capable of providing a robust real-time monocular SLAM solution over long distance even in relatively low frame rate. In [99], Strasdat, Montiel, and Davison compared the performance of the Bayesian filter based methods and the batch optimization (e.g., bundle adjustment) based methods. It has been concluded that if the computation resource is extremely limited, filter based approaches has advantages, otherwise the performance of optimization based ones is better. It is also verified that the number of tracked features in each frame is more important than the number of frames included in the estimation.

The batch optimization problem in monocular visual SLAM can be equivalently described by a graph model, which adapts to the long-researched graph-based SLAM framework from the robotics community, e.g., the GraphSLAM algorithm [73]. A review and tutorial of the graph-based SLAM algorithms is written by Grisetti et al. [100]. For the specific monocular VSLAM problem, the graph nodes have some sparse structure since the features are usually only visible to a few frames. Applying such sparse structures, Kaess proposed the state-of-the-art iSAM algorithm in [101], so that the optimization can be solved incrementally in real-time by using a factor graph model. Based on the work, Kaess et al. proposed an improved iSAM2 in [49] by applying a more advanced Bayesian tree graphic model, which showed improvement in both speed and accuracy. Another widely used optimization tool for monocular SLAM is g2o [48], which follows a similar idea that exploiting the sparsity of the problem to accelerate the least squares optimization.

As introduced earlier, a crucial problem of the monocular vision is that the scale of the translation is unavailable in the motion estimation of consecutive frames. Though the relative scale s_r between two motion steps can be estimated, the estimation is erroneous in practice due to the existence of noise. According to Eqn. (4.14), the error in relative scale estimation propagates and accumulates over time in a product manner. Consequently, the error in absolute scale increases exponentially, which is known as the scale drift problem. Strasdat, Montiel and Davison proposed a scale-aware loop closure method to deal with the relative scene scale drift [102] while correcting the accumulated error in motion estimation. The scale drift mitigation technique is applied in the state-of-the-art ORB-SLAM method [98]. Engel, Schöps and Cremers proposed a large scale dense SLAM (LSD-SLAM) algorithm for monocular cameras using pixel intensity values directly as measurements [33], which can also correct scale drifts over scenes when loop closures are detected. All the aforementioned approaches require loop closure to mitigate the drift. However, revisiting mapped places for loop closure may significantly reduce the task efficiency in an exploration mission. Moreover, the global scale problem still remains. All these algorithms estimate the motion and map points position only up to a global scale s_g.

A number of approaches have been considered for resolving the global scale ambiguity. Many of them use IMUs, see for example Achtelik et al. [103] and Nützi et al. [104]. However, the translation estimation from inertial sensors are calculated by two integrals from acceleration measurements. The error between two keyframes can grow large if the motion is slow (it would take long time to establish another keyframe since the scenes are varying slowly). Moreover, the sensor fusion of IMU and monocular camera is still a solution based on a dead-reckoning system. Tabibiazar and Basir proposed an approach using range measurements from cellular networks [105]. Their method requires at least 3 ranging links from known anchor locations for estimating the robot's position before integrating the result with vision-based estimates (loose coupling). Three anchors are often not available in Mars exploration mission due to the lack of infrastructure. Therefore, we developed a method in Section 4.2 for estimating the global scale in monocular visual SLAM using sparse range measurements from a single ranging link. In the case of a swarm of robots exploring Mars, these measurements could be performed between a single element at rest and all other elements [5]. Strictly, the method does not depend on the method of ranging, as long as it is performed with respect to a given location.

4.2 Global Scale Estimation using a Ranging Radio Link

As introduced in Section 4.1, navigation using a monocular camera has unobservability problem in states estimation. Besides the initial pose of the rover in the global frame (W), the global scale of the whole system is not uniquely resolvable given the visual measurements. We will show in this section that for dynamic rover in planar motion, by fusing sparse ranging measurements from a single radio link to an anchor with a monocular camera, the global scale can be estimated. At the same time, the unobservable state space in the global frame is reduced to only one dimensional as a polar angle, even if no initial pose is provided. The anchor point can be either a base station, or another rover in static mode in the swarm.

Fig. 4.5 illustrates the basic geometry between the static anchor and the dynamic rover in the motion plane. Without loss of generality, the reference frame origin is set to be the position of the base station. The direction of the x-axis of the reference frame (W) can be arbitrarily chosen. The initial heading of the camera is defined as the y'-axis in the navigation frame (N). In order to improve the convergence of the estimation, polar coordinates are used instead of Cartesian coordinates for position parameterizations. The detailed explanation will be provided when we introduce the fusion method. As a result, the initial 2D position of the rover $\vec{\beta}_{[1]}^{(W)}$ can be parameterized by the initial radius $r_{[1]}$ and the initial polar angle α in the world frame (W). The initial attitude of the rover is described by the angle $\alpha + \theta - \frac{\pi}{2}$, i.e.,

$$\vec{\beta}_{[1]}^{(W)} = \vec{t}_{(N \to W)} = r_{[1]} R(\alpha)[1, 0]^T, \tag{4.17}$$

$$R_{(1 \to W)} = R_{(N \to W)} = R(\alpha) R(\theta - \frac{\pi}{2}) = R(\alpha + \theta - \frac{\pi}{2}), \tag{4.18}$$

where $R(\cdot) \in \mathbf{SO}(2)$ is a 2D rotation matrix.

Using the images from the monocular cameras, the ego-motion of the rover in its navigation frame (N) can be independently estimated up-to-scale as $\{\hat{\beta}_{[k]}^{(N)}\}$ by exploiting a VSLAM algorithm. The time index k is defined as the discrete keyframe numbers. The keyframe selection criterion was introduced in Section 3.2.

In the world reference frame (W), the position of the rover at time k can be expressed in polar coordinates as

$$\vec{\beta}_{[k]}^{(W)} = s_g R_{(N \to W)} \vec{\beta}_{[k]}^{(N)} + \vec{t}_{(N \to W)}. \tag{4.19}$$

As a result, the coordinates of the trajectory in world frame and navigation frame can be related by a similarity transformation $T \in \mathfrak{Sim}(2) \subset \mathbb{R}^{3 \times 3}$. The transformation in Eqn. (4.19) can be parameterized by 4 parameters as:

$$\vec{\beta}_{[k]}^{(W)} = R(\alpha) \left(r_{[1]} [1, 0]^T + s_g R(\theta - \frac{\pi}{2}) \vec{\beta}_{[k]}^{(N)} \right). \tag{4.20}$$

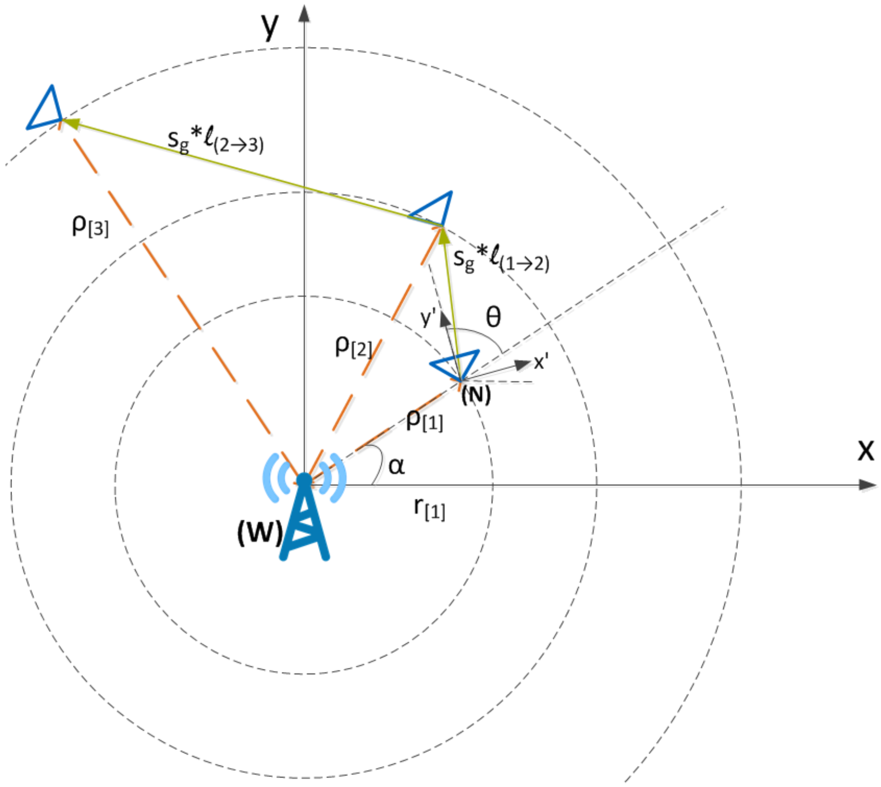


Figure 4.5: Geometry of the static station and the dynamic rover

Although the monocular camera itself can only estimate the motion up-to-scale, with the additional help of a sparse set of noisy range measurements $\{\rho_{[k]}\}$ obtained at the time instant k, where

$$\rho_{[k]} = r_{[k]} + \eta_{[k]} = \left\| \vec{\beta}_{[k]}^{(W)} \right\| + \eta_{[k]}, \tag{4.21}$$

a method for estimating the global scaling factor s_g can be devised by exploiting ranging measured at keyframes. It is assumed here that the range measurements has been corrected with the height difference between the base station transmission antenna and the receiver antenna on the rover, so that the measurements are projected to the 2D motion plane.

After obtaining N_k keyframes, by fusing the camera-based trajectory estimates with the range measurements, the global scale s_g, initial attitude heading θ, and initial radius $r_{[1]}$ can be estimated by least-squares optimization. Stacking the N_k range measurements and the three parameters into vectors $\rho = [\rho_{[1]}, \rho_{[2]}, ..., \rho_{[N_k]}]^T$ and $F(\xi) = [\|\vec{\beta}_{[1]}^{(W)}\|, \|\vec{\beta}_{[2]}^{(W)}\|, ..., \|\vec{\beta}_{[N_k]}^{(W)}\|]^T$ with $\xi = [s_g, \theta, r_{[1]}]^T$, the problem can be formulated as

$$\hat{\xi} = \arg\min_{\xi} \|\rho - F(\xi)\|_{Q^{-1}}^2 \qquad \text{s.t. } B\xi > 0. \tag{4.22}$$

The inequality constraints are due to the positiveness of both the scale s_g and the initial true range $r_{[1]}$. $B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is a selection matrix used to set the constraints. Q is the covariance matrix of the noise $\eta = [\eta_{[1]}, \eta_{[2]}, ..., \eta_{[N_k]}]^T$.

Due to the presence of several local minima and the bounded search space, it is challenging to solve the nonlinear inequality constrained optimization in Eqn. (4.22). However, not all minima violating the constraints represent erroneous solution, due to the symmetric properties of the objective function. Define $A_k(s_g, \theta, r_{[1]}) = (r_{[k]}(\xi) - \rho_{[k]})^2$. For any s_g, θ and $r_{[1]}$, the value of object function is invariant to the

Table 4.1: Transformation on the results from unconstrained optimization.

If		Transformation	
$\hat{s}_g < 0$	$\hat{r}_{[1]} > 0$	$\hat{s}_g \leftarrow -\hat{s}_g$	$\hat{\theta} \leftarrow \hat{\theta} + \pi$
$\hat{s}_g > 0$	$\hat{r}_{[1]} < 0$	$\hat{r}_{[1]} \leftarrow -\hat{r}_{[1]}$	$\hat{\theta} \leftarrow \hat{\theta} + \pi$
$\hat{s}_g < 0$	$\hat{r}_{[1]} < 0$	$\hat{s}_g \leftarrow -\hat{s}_g$	$\hat{r}_{[1]} \leftarrow -\hat{r}_{[1]}$

following parameter change:

$$\begin{aligned} A_k(s_g, \theta, r_{[1]}) &= A_k(-s_g, \theta + \pi, r_{[1]}) \\ =A_k(s_g, \theta + \pi, -r_{[1]}) &= A_k(-s_g, \theta, -r_{[1]}). \end{aligned} \tag{4.23}$$

Consequently, we can obtain the estimates of the parameters by solving the corresponding unconstraint problem and by transforming the results obtained with the relations given in Table 4.1.

The unconstrained optimization is still non-linear due to the property of function $F(\xi)$, so it is solved by linearizing at initial point $\hat{\xi}_1$ and updating the estimates iteratively. The convergence performance of the optimization depends on the accuracy of the initial values. Estimating the scale s_g is not significantly sensitive to the initialization, since the ranging measurements constrain the scale of the trajectory to certain extent. Moreover, for the reason that the ranging measurement $\rho_{[1]}$ is a precise approximation of $r_{[1]}$, the most crucial factor is choosing the initial value of the heading angle θ. In practice, we initialize θ with multiple hypothesis uniformly distributed in $[0, 2\pi)$, and choose the result with smallest residual. Utilizing the ranging measurement as initial value for $\rho_{[1]}$ is the main motivation of parameterizing the position by polar coordinates. For a Cartesian coordinates parameterization, the initial value of the position has to be searched in parallel in a two dimensional space. As a result, the solution of Eqn. (4.22) can be obtained by iteratively solving the linearized problem as

$$\hat{\xi} = \arg\min_{\xi} \|\rho - J_\xi \xi\|^2_{Q^{-1}} \tag{4.24}$$

$$\hat{\xi}_{i+1} = \hat{\xi}_i + \left(J_\xi(\hat{\xi}_i)^T Q^{-1} J_\xi(\hat{\xi}_i)\right)^{-1} J_\xi(\hat{\xi}_i)^T Q^{-1} \left(\rho - F(\hat{\xi}_i)\right) \tag{4.25}$$

where J_ξ is the Jacobian matrix associated to the function $F(\xi)$.

In order to solve the problem in Eq. (4.24), $K \geq 3$ range measurements are required. Due to the high nonlinearity of the objective function, the Levenberg-Marquardt algorithm is applied, instead of a Gauss-Newton approach, in order to exploit its superior global minimization capabilities.

As a result, the global scale factor s_g is estimated by combining ranging and visual measurements. Consequently, the pose of the rover and the coordinates of the map points are estimated without scale ambiguity. In addition, the initial heading of the rover θ, which refers to the radial direction between the rover and the station, can be obtained from the estimation. It should be mentioned that the ranges are invariant to the change of the initial polar angle α. Hence with a single radio link, the absolute position of the rover in reference frame (W) is ambiguous by the angle. The ambiguity can be resolved only if additional set-ups are available, e.g., by adding the second station, or by using an antenna array to estimate the angle of arrival.

The proposed scale estimation method using sparse range measurements is tested in simulation using KITTI benchmark datasets [106]. We use the odometry dataset with provided ground truth to verify the result. The range measurements $\{\rho_{[k]}\}$ are simulated from the true ranges to anchor point located at $\vec{0}$ with additive Gaussian noise.

Fig. 4.6 shows the scale estimation result until keyframe 100 from KITTI odometry dataset 07. The uncertainty of the range measurements is 1 [m]. The global scale s_g is initialized arbitrarily (initialized as $s_g = 1$ in the simulation), which results in large initial error. With increasing number of measurements from the camera and the ranging signal, the error of the estimated global scale declines quickly, and remains

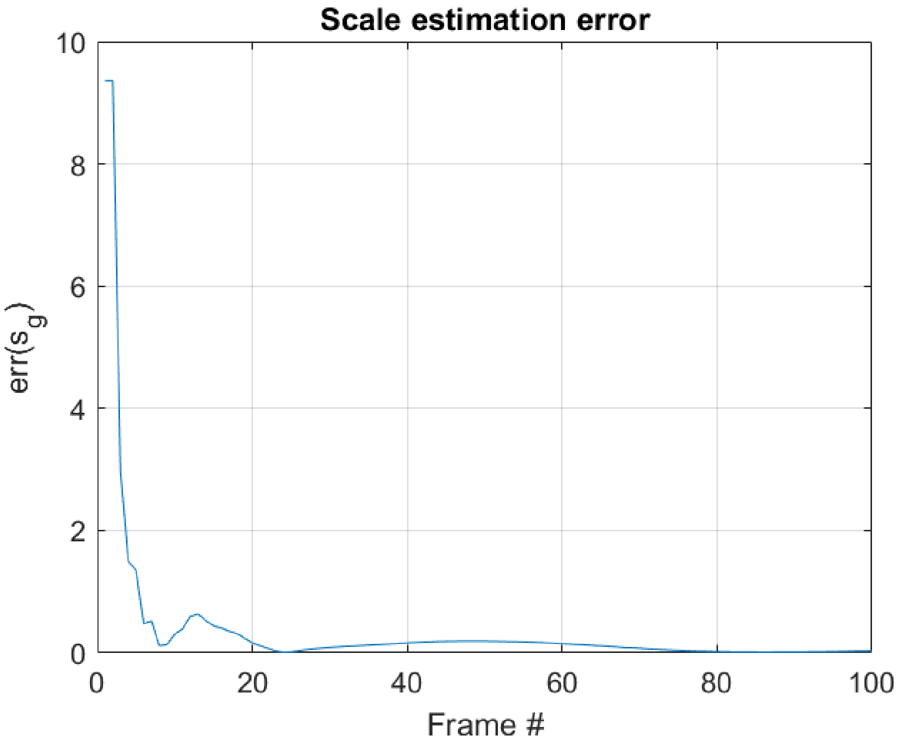
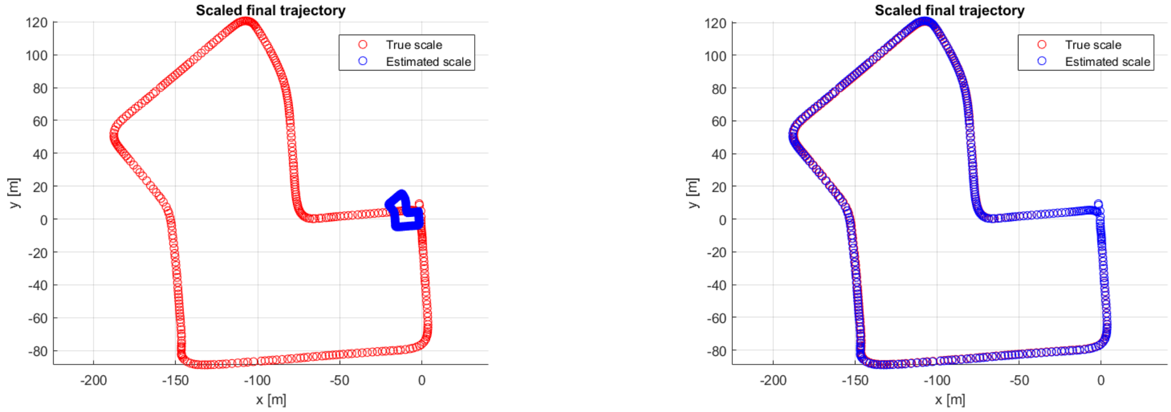


Figure 4.6: Scale estimation using KITTI odometry dataset 07



(a) Frame 2

(b) Frame 100

Figure 4.7: Estimated trajectory applying true global scale and the estimated scale for KITTI odometry dataset 07

low after 20 keyframes. As a straightforward comparison, Fig. 4.7 illustrates the difference between the estimated full trajectory of the dataset using true global scale and estimated scale respectively. The red trajectory represents the result scaled by the true global scale, and the blue one shows the trajectory using the estimated scale. Fig. 4.7a illustrates result using the initial guess without any scale estimation. It can be seen that with an arbitrarily initialized value, the global scale is significantly erroneous. Fig. 4.7b shows the results using the estimated scale value at keyframe 100. At that moment, the estimated result almost align with the ground truth. Therefore, the global scale can be reliably estimated using the proposed method.

To analyze the impact of the ranging noise on the estimation result, we execute the simulation by adding different ranging noise on the measurements. Fig. 4.8 visualizes the result of the global scale estimation error with respect to the ranging noise level. The images used in the simulation are taken from KITTI dataset 07. The curve shows the root-mean-square error (RMSE) of global scale estimation with respect to the

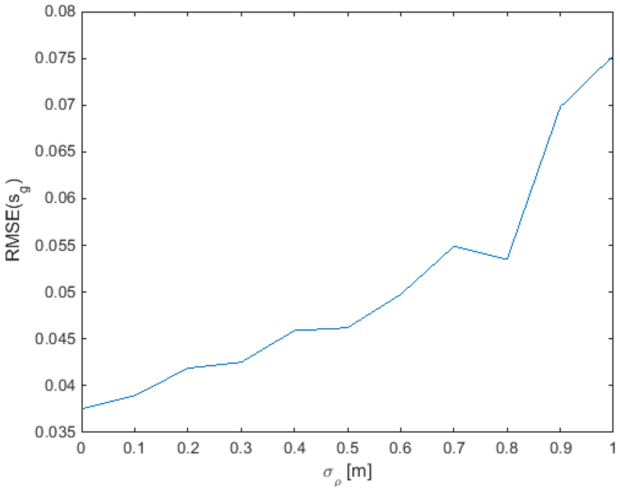


Figure 4.8: Impact of ranging error on the scale estimation error

standard deviation of the range measurements. It can be seen that the scale estimation error is superlinearly dependent on the range measurements error. The curve is generated using KITTI dataset 07 with ground truth scale $s_g = 10.3624$. The relative error is less than 0.8% when the ranging uncertainty is below 1 meter. In practice, dependent on the requirements and constraints for the SLAM scenario, higher ranging accuracy than 1 meter is feasible through signal design.

Fig. 4.9, Fig. 4.10 and Fig. 4.11 shows the estimated poses of the rover using images from KITTI odometry dataset 04, 03 and 07 by applying the jointly estimated parameters $\hat{\xi}$. The dataset 04 contains images taken from a linear trajectory without any direction change. The three datasets represent different typical motions of a dynamic rover. The dataset 03 is a forward trajectory with a few turns, and the dataset 07 consists of images taken from motions in a closed loop. In the figures, the red trajectories are the ground truth, and the blue ones are the estimated trajectories. In the simulation, the standard deviation of the ranging noise is 1 meter. It can be conclude from the results that using the proposed method, the poses of the rover can be well estimated with only a single camera and sparse ranging measurements from a single base station.

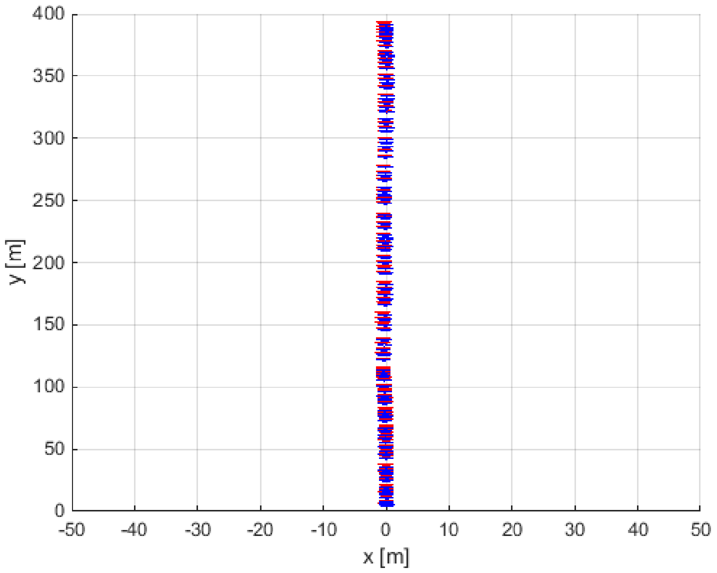


Figure 4.9: Pose estimation using KITTI odometry dataset 04, σ_ρ=1 m

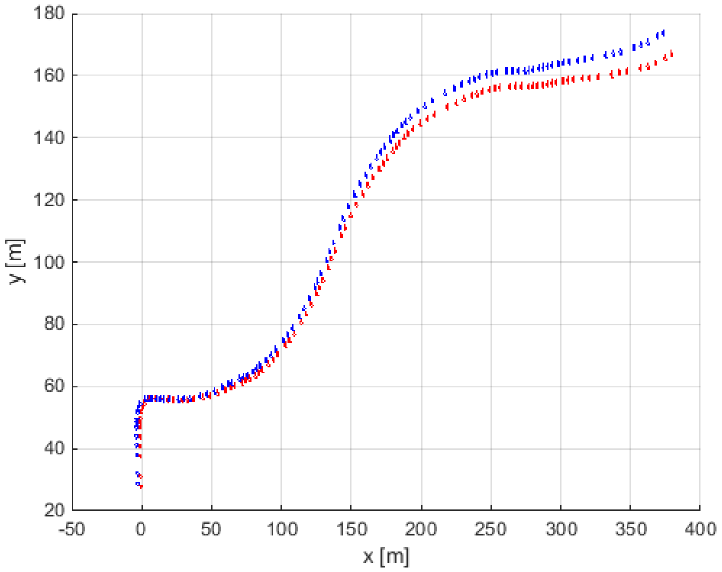


Figure 4.10: Pose estimation using KITTI odometry dataset 03, σ_ρ=1 m

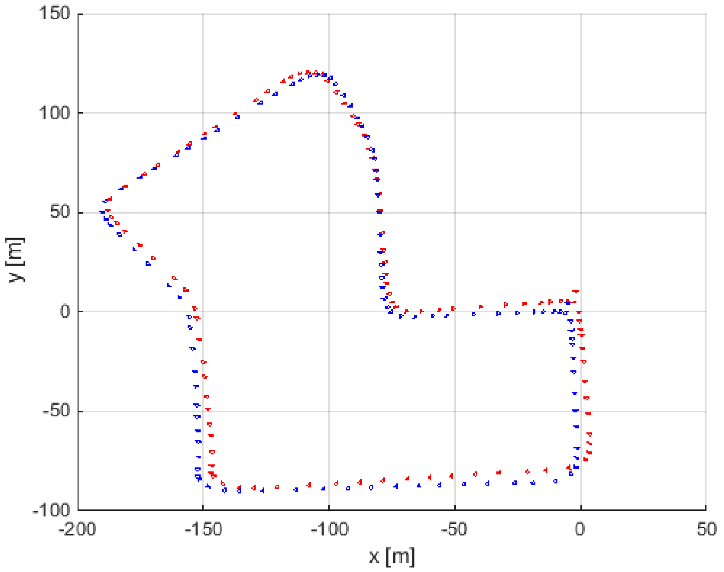


Figure 4.11: Pose estimation using KITTI odometry dataset 07, σ_ρ=1 m

4.3 Determine the Ambiguous Polar Angle

In the above section, we developed a sensor fusion method to estimate the global scale in monocular VSLAM by exploiting range measurements from a single radio link. In the parameter estimation, the scale factor s_g is coupled with other geometric parameters of the relative pose between the rover and the base station. As a result, the initial heading angle θ, and the initial relative distance $r_{[1]}$ are jointly estimated with the global scale. However, using the monocular camera along with the range measurements from a single base station, the relative polar angle α cannot be determined without ambiguity. In this section, we discuss the possibility to resolve the polar angle ambiguity with additional information.

The first option is to use an antenna array instead of a single antenna for the onboard radio receiver. Exploiting the phase difference of the received signal from the antennas in the array, the direction of arrival (DoA) of the radio signal can be estimated by algorithm such as MUSIC [107].

When an antenna array is unavailable, the polar angle estimation needs to rely on a second base station, which can either be another static rover in the swarm or be some infrastructure with known position. The coordinates of the two stations in the global frame (W) are assumed to be known. Here we discuss two scenarios:

1) The rover can connect to both base stations simultaneously at some instant;
2) The rover connects to a second base station shortly after it loses connection to the first station.

Fig. 4.12 illustrates the first scenario: a mobile rover enters a zone with connectivity to two static base stations. The two stations can be either infrastructures or two other rovers on static mode in a robotic swarm. In such case, the model of the geometric parameters is shown in Fig. 4.13. As the single station case, we choose the position of the first base station as the reference frame (W). The navigation frame (N) of the rover is related to (W) by a transformation which can be parameterized by radius $r_{[1]}$, polar angle α and heading angle θ. Using the method from Section 4.2, the rover can exploit its egomotion estimation in navigation frame from the camera $\{\hat{\beta}^{(N)}_{[k]}\}$ and the range measurements from the radio link $\{\rho_{[k]}\}$ to estimate all the parameters except α. By adding the ranging measurements from the second base station $\{\rho'_{[k]}\}$, the 2D position of the rover in the world frame can be solved with a mirroring ambiguity at any time instant k by

$$\hat{\beta}^{(W)}_{[k]} = \underset{\vec{\beta}^{(W)}_k}{\arg\min} \frac{1}{\sigma^2_{[k]}} \left(\rho_{[k]} - \|\vec{\beta}^{(W)}_k - \vec{x}^{(W)}_b\|\right)^2 + \frac{1}{\sigma'^2_{[k]}} \left(\rho'_{[k]} - \|\vec{\beta}^{(W)}_k - \vec{x}^{(W)}_{b'}\|\right)^2, \tag{4.26}$$

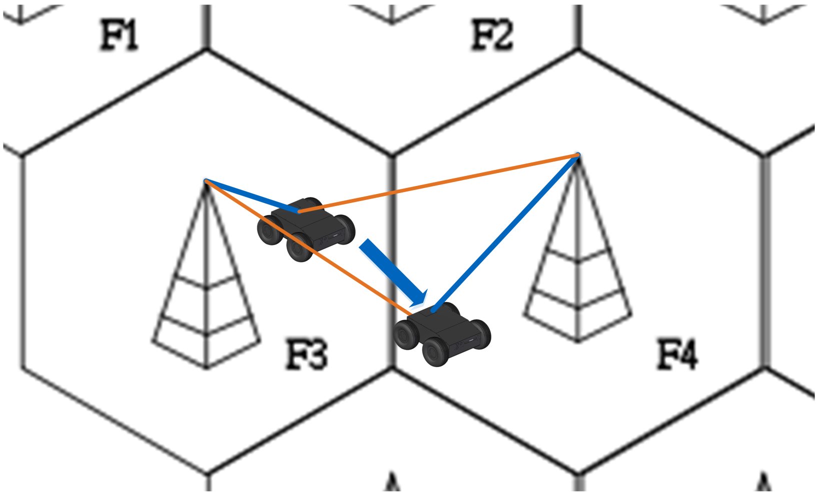


Figure 4.12: Connect to two base stations simultaneously

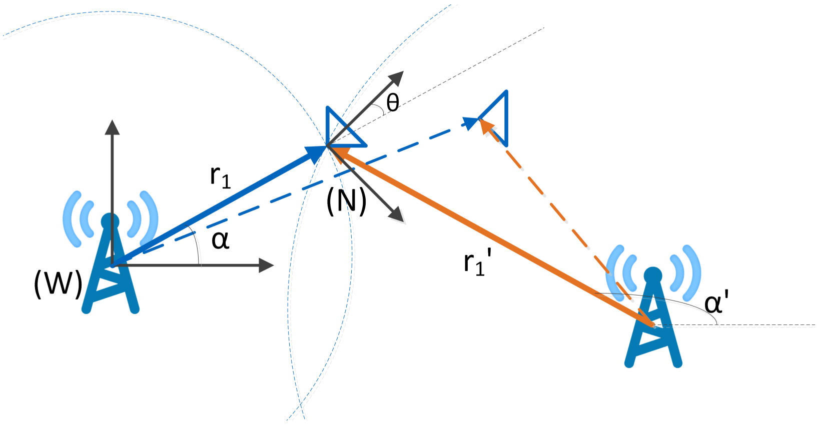


Figure 4.13: Model for two station scenario

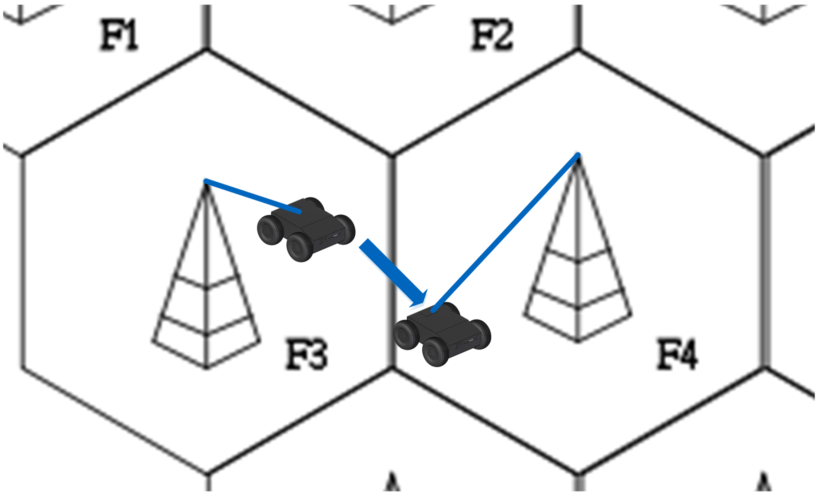


Figure 4.14: Connect to a different base station after losing the connection to the previous one

where $\vec{x}^{(W)}_{b'}$ is the position of the second base station. The two solutions of the optimization in Eqn. (4.26) reduce the ambiguity of the polar angle α to two possibilities, which corresponds to the heading angle value θ and $-\theta$, respectively. Consequently, with the estimated $\hat{\theta}$, the mirroring ambiguity can be resolved and the whole geometry can be estimated without any ambiguity. The positioning accuracy can be improved either by Bayesian filter based tracking or by batch optimization using measurements at multiple epochs as:

$$\{\hat{\beta}^{(W)}_{[k]}\} = \underset{\{\vec{\beta}^{(W)}_{k}\}}{\arg\min} \left\|\rho - F(\{\vec{\beta}^{(W)}_{k}\})\right\|^2_{Q^{-1}} + \left\|\rho' - F'(\{\vec{\beta}^{(W)}_{k}\})\right\|^2_{Q'^{-1}}, \tag{4.27}$$

where $\rho' = [\rho'_{[1]}, \rho'_{[2]}, ..., \rho'_{[N_k]}]^T$ are the range measurements from the second station and $F'(\{\vec{\beta}^{(W)}_{[k]}\}) = [\|\vec{\beta}^{(W)}_{[1]} - \vec{x}^{(W)}_{b'}\|, \|\vec{\beta}^{(W)}_{[2]} - \vec{x}^{(W)}_{b'}\|, ..., \|\vec{\beta}^{(W)}_{[N_k]} - \vec{x}^{(W)}_{b'}\|]^T$ describes the distance from the rover to the second base station with position. Q and Q' are the covariance matrices of the ranging noise from the two radio links respectively.

This scenario is actually a simplified situation of the single station case in Eqn. (4.22). However, due to coverage problem, the rover cannot ensure consistent simultaneous connectivity to two stations. As a less demanding scenario, Fig. 4.14 illustrates the situation that the rover connect to another station after it loses the first station's link during the motion for a while. Under such circumstance, the polar angle α can also be determined without ambiguity.

During the whole handover process, the rover can estimate its egomotion (up to global scale) in its navigation frame (N) using the onboard monocular camera. Denote the position in the navigation frame as $\{\hat{\beta}^{(N)}_{[k_1]}|k_1 = 1...N_{k1}\}$, $\{\hat{\beta}^{(N)}_{[T+k_2]}|k_2 = 1...N_{k2}\}$ and $\{\hat{\beta}^{(N)}_{[\tau]}|\tau = N_{k1} + 1...T\}$ respectively for the periods of time that the rover connects to the first station, the second station, and without any connection in between.

For each of the stations, the beacon location is the navigation frame of the rover can be parameterized using range at the frame and azimuth angle φ, as shown in Fig. 4.15. The beacon direction azimuth angle φ is an equivalent parameterization of the heading angle θ in the navigation frame. The transformation between the two representations in Fig. 4.5 and Fig. 4.15 follows the relation

$$\varphi = \frac{\pi}{2} - \theta + \pi = \frac{3\pi}{2} - \theta. \tag{4.28}$$

Since the heading angle of the first keyframe with connectivity can be estimated using the method in Section 4.2, the location of the two stations in the navigation frame, can be calculated as $\hat{x}^{(N)}_{b}$ and $\hat{x}^{(N)}_{b'}$, according to the estimated global scale and heading angle. If the position of both stations in the global frame is known,

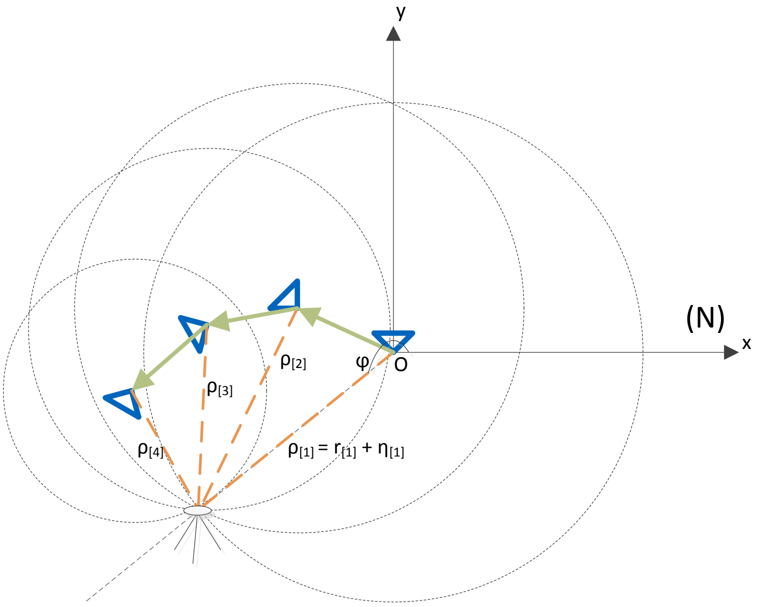


Figure 4.15: Beacon location in the navigation frame.

the positioning problem in the global reference frame (W) is transferred to find the optimal similarity transformation between the two reference frames. In the planar motion case, the problem is simplified to the following least-squares estimation:

$$\{\hat{t}_{(N\to W)}, \hat{R}_{(N\to W)}, \hat{s}_{(N\to W)}\} = \arg\min_{t,R,s} \sum_{i=1}^{N_b} \|sR\hat{x}_{b_i}^{(N)} + t - \vec{x}_{b_i}^{(W)}\|^2, \tag{4.29}$$

where $t \in \mathbb{R}^2, R \in \mathbf{SO}(2), s \in \mathbb{R}$, and N_b is the number of the reference points used to match the two reference frames. For this two-station problem, $N_b = 2$.

If the transformation is constrained to be rigid transformation, i.e., $s = 1$, the problem is simplified to the well-known Wahba's problem [108]. The problem can be solved by Kabsch algorithm [109], which calculate the estimated rotation using singular value decomposition (SVD). Based on the Kabsch algorithm, the optimal transformation scale can be estimated as

$$\hat{s}_{(N\to W)} = \frac{\max\{S_{WW}\}}{\max\{S_{NW}\}}, \tag{4.30}$$

where $\{S_{WW}\}$ and $\{S_{NW}\}$ are the singular values of the covariance matrices $\vec{x}_b^{(W)}(\vec{x}_b^{(W)})^T + \vec{x}_{b'}^{(W)}(\vec{x}_{b'}^{(W)})^T$ and $\hat{x}_b^{(N)}(\vec{x}_b^{(W)})^T + \hat{x}_{b'}^{(N)}(\vec{x}_{b'}^{(W)})^T$, respectively.

By knowing the coordinate transformation between the world frame and the navigation frame, the rover trajectory can be transformed to estimates in the world frame without any ambiguity as:

$$\hat{\beta}_{[k]}^{(W)} = \hat{s}_{(N\to W)}\hat{R}_{(N\to W)}\hat{\beta}_{[k]}^{(N)} + \hat{t}_{(N\to W)}. \tag{4.31}$$

In addition, the attitude in the global reference frame can be estimated as:

$$\hat{R}_{(k\to W)} = \hat{R}_{(N\to W)}\hat{R}_{(k\to N)}, \tag{4.32}$$

where $\hat{R}_{(k\to N)}$ can be obtained from visual SLAM estimates.

As a result, using the ranging measurements from a second static base station, the rover pose relative to the base stations can be estimated without ambiguity.

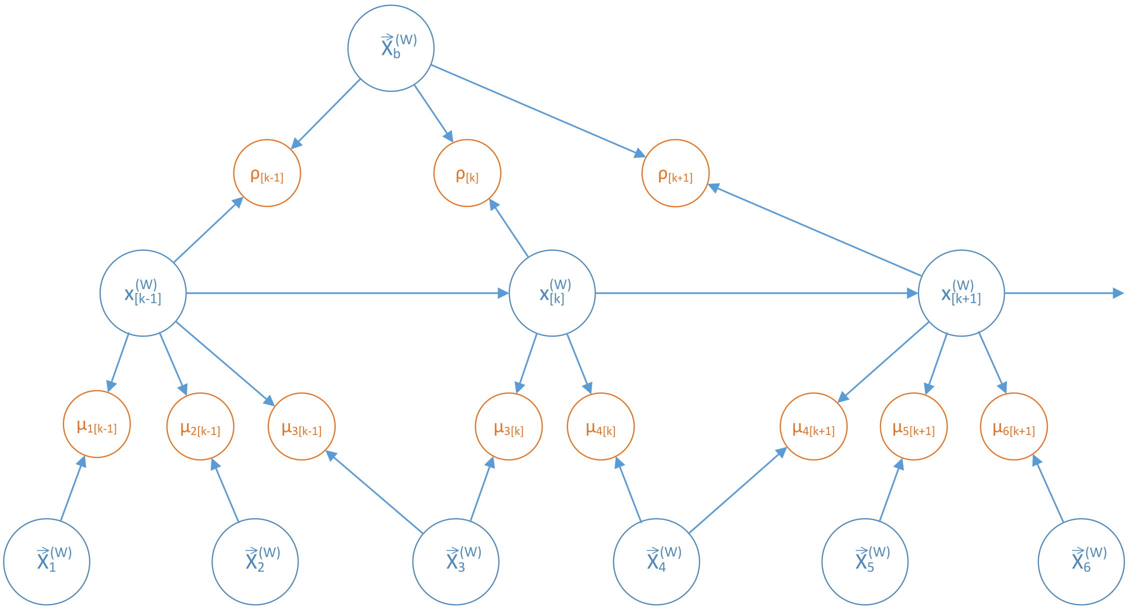


Figure 4.16: Bayesian network expression of camera and ranging sensor fusion

4.4 Precise 2D Visual SLAM using Monocular Camera and Ranging Fusion

In the above methods which estimate the global scale and the rover pose, the visual and ranging measurements from the two sensors are fused in a loose coupling way, i.e., the motion is estimated using the camera only and then coupled with the range measurements for sensor fusion. In this section, we move a step further towards tight coupling of the two sensors, so that the pose estimation from the VSLAM can be improved by exploiting the ranging measurements.

As introduced in Section 3.2 and Section 4.1, the visual odometry technique is a navigation method based on dead reckoning concept. For the reason that the error of motion estimation accumulates over time, the estimated rover pose using camera only will drift from the true value as the rover moves. If the camera can observe and detect some features which are already in the built map, the drift of the whole trajectory can be mitigated using loop closure techniques, since the uncertainty of the prior probability of the re-observed feature location is significantly reduced. However, revisiting a mapped place is not feasible in many applications, which results in consistently increasing estimation uncertainty due to the lack of loop closure. As a solution, a static base station can provide an anchor point to the rover, since the noise of the ranging measurements can be assumed to be independent over time. By exploiting the ranging measurements from the station and tightly coupling the sensors, the drift of the rover pose estimation can be mitigated without revisiting mapped places.

By multiplying the estimated global scale to the camera-based position estimates, a set of camera coordinates in the global frame metric can be obtained as $\{\hat{s}_g \hat{\beta}_{[k]}^{(N)} | k = 1, ..., N_k\}$. The set of positions is a coarse estimate of $\{\vec{\beta}_k^{(W)} | k = 1, ..., N_k\}$ (with a polar angle ambiguity α which can be potentially resolved using the methods in Section 4.3). The coarse estimates can be used as initial values to fine tune the pose estimation by applying a batch processing of the state variables stacked over time. The optimization is executed on the positions in the global frame (W), so that the error in scale $\hat{s}_g$ and in monocular egomotion $\hat{\beta}_{[k]}^{(N)}$ do not need to be considered separately. It should be mentioned that the following tight coupling fusion can still improve the trajectory estimation accuracy if the polar angle α is not resolvable.

Fig. 4.16 illustrates the Bayesian network for the fusion of the visual features from the camera and ranging measurements from the wireless radio link. Since control is not the main concern of our work, without loss of generality, the known control input is ignored in the Bayesian network.

The joint probability of the variables in the graph is

$$p_{joint} = \prod_{i=1}^{N_p} \prod_{k=1}^{N_k} p(x_{[0]}^{(W)}, x_{[1]}^{(W)}, ..., x_{[N_k]}^{(W)}, \vec{X}_1^{(W)}, ..., \vec{X}_{N_p}^{(W)}, \vec{X}_b^{(W)}, \mu_{1[1]}, ..., \mu_{N_p[N_k]}, \rho_{[1]}, ..., \rho_{[N_k]}), \tag{4.33}$$

where $x_{[k]}^{(W)}$ denotes the pose of the rover in global frame at time k, $\vec{X}_b^{(W)}$ the anchor point position, $\vec{X}_i^{(W)}$ the i-th feature position. $\rho_{[k]}$ and $\mu_{i[k]}$ are the corresponding ranging and visual measurements respectively.

In the sensor fusion, the noise of the feature location is independent of the ranging noise. Exploiting the independency of the measurements, the joint probability can be factorized as

$$p_{joint} = \prod_{i=1}^{N_p} \prod_{k=1}^{N_k} p(\mu_{i[k]} | \vec{X}_i^{(W)}, x_{[k]}^{(W)})^{v_{i[k]}} p(\vec{X}_i^{(W)}) p(x_{[0]}^{(W)}) p(x_{[k]}^{(W)} | x_{[k-1]}^{(W)}) p(\rho_{[k]} | x_{[k]}^{(W)}, \vec{X}_b^{(W)}) p(\vec{X}_b^{(W)}), \tag{4.34}$$

where $v_{i[k]}$ is the binary visibility masking that $v_{i[k]} = 1$ if the feature i is visible in keyframe k.

The SLAM task using the camera-ranging fusion can be formulated as the following MAP estimator:

$$\{\hat{x}_{[k]}^{(W)}, \hat{X}_i^{(W)}\} = \underset{\{x_{[k]}^{(W)}, \vec{X}_i^{(W)}\}}{\arg\max} \; p(x_{[1]}^{(W)}, ..., x_{[N_k]}^{(W)}, \vec{X}_1^{(W)}, ... \vec{X}_{N_p}^{(W)} | \vec{X}_b^{(W)}, \mu_{1[1]}, ..., \mu_{N_p[N_k]}, \rho_1, ..., \rho_{N_k}). \tag{4.35}$$

The a priori distribution of the initial states $x_{[0]}^{(W)}$ and the base station position $\vec{X}_b^{(W)}$ in the reference frame can be estimated using the method in Section 4.2 and Section 4.3. Moreover, the a priori distribution of the feature points $p(\vec{X}_i^{(W)})$ is unknown in exploration missions. Therefore, using the factorization in Eqn. (4.34) and a specific state transition model, the optimal solution of the MAP estimator given the available knowledge can be estimated by:

$$\{\hat{x}_{[k]}^{(W)}, \hat{X}_i^{(W)}\} = \underset{\{x_{[k]}^{(W)}, \vec{X}_i^{(W)}\}}{\arg\max} \prod_{i=1}^{N_p} \prod_{k=1}^{N_k} p(\mu_{i[k]} | \vec{X}_i^{(W)}, x_{[k]}^{(W)})^{v_{i[k]}} p(\rho_{[k]} | x_{[k]}^{(W)}, \vec{X}_b^{(W)}) p(x_{[k]}^{(W)} | x_{[k-1]}^{(W)}). \tag{4.36}$$

Without a proper motion model, the process noise of the state transition model has large covariance. As a result, the motion model makes the pose estimation more robust against sensor data outages, but the impact of the motion model can be ignored if the sensor measurements are consistently reliable. In such conditions, the MAP estimator is approximately equivalent to the following log-likelihood estimator:

$$\{\hat{x}_{[k]}^{(W)}, \hat{X}_i^{(W)}\} = \underset{\{x_{[k]}^{(W)}, \vec{X}_i^{(W)}\}}{\arg\max} \sum_{i=1}^{N_p} \sum_{k=1}^{N_k} v_{i[k]} \log p(\mu_{i[k]} | \vec{X}_i^{(W)}, x_{[k]}^{(W)}) + \sum_{k=1}^{N_k} \log p(\rho_{[k]} | x_{[k]}^{(W)}, \vec{X}_b^{(W)}). \tag{4.37}$$

Under Gaussian noise assumption, the log-likelihood estimator can be transformed to the following least-squares optimization

$$\{\hat{x}_{[k]}^{(W)}, \hat{X}_i^{(W)}\} = \underset{\{x_{[k]}^{(W)}, \vec{X}_i^{(W)}\}}{\arg\min} \sum_{i=1}^{N_p} \sum_{k=1}^{N_k} v_{i[k]} \| \mu_{i[k]} - \pi(\vec{X}_i^{(W)}, x_{[k]}^{(W)}) \|_{\Sigma_u^{-1}}^2 + \sum_{k=1}^{N_k} \| \rho_{[k]} - F(x_{[k]}^{(W)}, \vec{X}_b^{(W)}) \|_{\Sigma_\rho^{-1}}^2, \tag{4.38}$$

where $\pi(\cdot)$ and $F(\cdot)$ is the projection function and the range function defined in Section 2.2 and 2.4 respectively. Obviously, the both functions are non-linear with respect to the parameters to be estimated. Hence, initial values for starting the parameter optimization are required. The coarse estimation of the camera poses and the map point location is available from monocular visual SLAM output, while the global scale as well as the relative pose between the rover and the static station can be obtained from the loose coupling fusion that we proposed in Section 4.4 and 4.3.

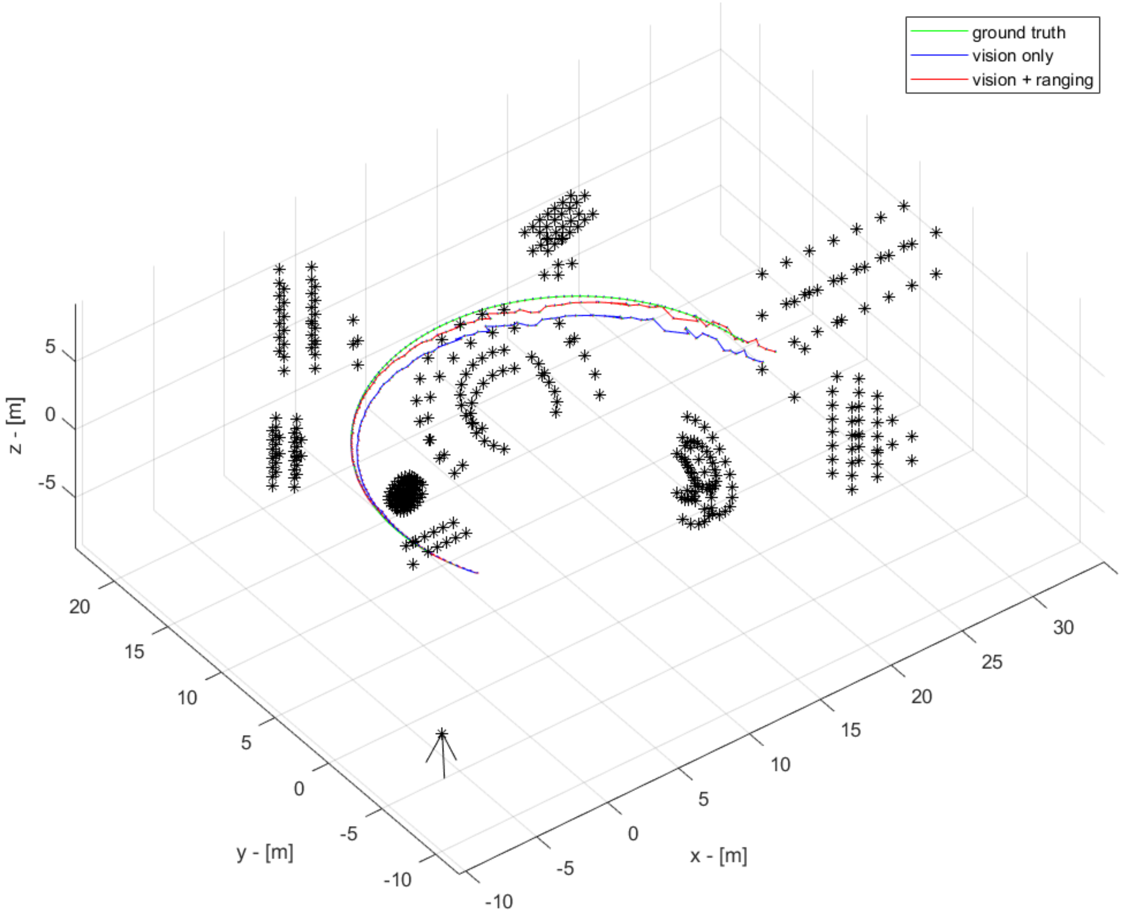


Figure 4.17: Trajectory estimation using camera and ranging sensor fusion, $\sigma_\rho = 0.1$ m

Compared with the camera only bundle adjustment approach in Eqn. (3.13), the additional terms introduced by the range measurements in the batch least squares optimization in Eqn. (4.38) help reduce the drift of the dead-reckoning visual navigation system in the long run, because the static base station provides an anchor point so that the ranging error does not accumulate over time. As a result, the error of the trajectory and map points location estimation can be mitigated using the sensor fusion method without any requirement to revisit old places for global loop closure.

Unlike sensors such as lidar, which provide dense ranging measurements, another advantage of the fusion method is that the number of the ranging measurements is bounded by the number of keyframes and increases at most linearly with time (when the radio link is available at every keyframe). In addition, the range between the radio transmitter and the receiver is only a function of the rover position given a known base station location. Consequently, the sparsity of the object function is not affected by introducing the sparse range measurements, so that the smoothing algorithms such as iSAM2 [49] can be implemented without any barrier.

Fig. 4.17 shows an instance of the rover trajectory estimation using monocular camera and sparse ranging measurements from a single base station. The green curve is the true trajectory in the simulation, and the feature points are drawn as black stars in the 3D space. The optimization is executed using the GTSAM library [110]. In the simulation, the ranging noise standard deviation is 10 [cm], and the visual measurement noise is 0.3 [pixel] for both image dimensions. The parameters are reasonable empirical values according to our hardware measurements. The estimated trajectory using monocular VSLAM is plotted in blue (scaled with the true global scale), and the red trajectory represents the estimation result using both the visual measurements and the sparse ranging measurements from the static anchor point. It can be seen from

Table 4.2: Estimation error of the rover position.

	Vision-only	Vision-range-fusion
RMSE($\hat{x}$) - [m]	0.9178	0.6493

the simulation result that the vision-only approach drifts slowly away from the true trajectory due to error accumulation, even if the true global scale is assumed to be known. Meanwhile, the drift can be mitigated if the ranging measurements are included in the optimization. The improvement is significant if the rover travels for a long period of time without revisiting known places for loop closures. Table 4.2 shows the root-mean-square-error (RMSE) of the rover positions in the world frame for both methods.

5. Visual Navigation of a Vehicle Pair in Robotic Swarms

As shown in Chapter 4, the scale problem in monocular VSLAM can be solved by introducing a single radio link with ranging capability, e.g., from another static mode rover in the swarm. Moreover, the relative pose between the dynamic rover and the base station can be estimated with an azimuth angle ambiguity. Based on the initial coarse geometry estimation, the visual SLAM performance can be improved by tightly coupled sensor fusion.

Besides the ego-motion estimation, we are also interested in the relative pose between a pair of vehicles in swarm navigation. For instance, a rover exploring in caves (denoted as rover 1) may only have one radio connection to another rover outside (rover 2). The rover 2 can serve as a relay to exchange data between rover 1 and other vehicles in the swarm. At the same time, it can act as the static base station with ranging link, so that the scale estimation and sensor fusion methods proposed in Chapter 4 can be applied on rover 1. However, even if the relay rover is aware of its position in the global frame, the estimated trajectory of the exploring rover in the cave is still ambiguous due to the unobservable polar angle.

Compared with the case that a static rover serving as a base station, the dynamics of the second rover introduces more degrees of freedom to be estimated. Nevertheless, at the same time, the egomotion estimation capability of the rover also provides more information for estimating the geometric parameters. In this chapter, we show that the relative pose between two dynamic rovers can be estimated, if one of the rovers can transmit its local ego-motion estimates to the other one through the radio link. The approach does not demand to transmit any image or feature descriptor data, so in general the communication throughput requirements is feasible in practice. As a result, if one of the rover is aware of its pose in the global frame, another rover can also localize itself in the same reference frame according to the estimated relative pose.

5.1 Scale and 2D Relative Pose Estimation using Monocular Camera and Ranging Fusion

Without any other anchor point with known absolute position, one can only estimate the position and attitude of the cameras with respect to a known point in the navigation frame. We choose the initial position of the camera projection center of rover 2 as the coordinate reference system's origin, and the camera's principal axis as the y-axis. Fig. 5.1 illustrates the reference system and the geometry of the two rovers. The initial position and attitude of the two rovers can be expressed in the reference frame as

$$ {}^{\mathbf{1}}\vec{\beta}^{(W)}_{[1]} = r_{[1]} R(\alpha)[1,0]^T,\ R_{(N_1 \to W)} = R(\alpha + \theta - \frac{\pi}{2}). \tag{5.1}$$

$$ {}^{\mathbf{2}}\vec{\beta}^{(W)}_{[1]} = [0,0]^T, \quad R_{(N_2 \to W)} = I_2, \tag{5.2}$$

where I_2 denotes the two-dimensional identity matrix, and $R(\cdot) \in \mathbf{SO}(2)$ denotes a 2D rotation matrix.

Using the images from the monocular cameras, the egomotion of the two rovers in their navigation frames can be independently estimated up-to-scale as $\{{}^{\mathbf{1}}\vec{\beta}^{(N_1)}_{[k]}\}$ and $\{{}^{\mathbf{2}}\vec{\beta}^{(N_2)}_{[k]}\}$. The global scales of the two cameras are in general different, since we defined the global scale for a camera in Section 4.1 by assuming that the first translation distance in the navigation frame is unit, and the motion of the two rovers are independent. In the common reference frame (W), the position of the two rovers at k-th keyframe can be expressed as

$$ {}^{\mathbf{1}}\vec{\beta}^{(W)}_{[k]} = {}^{\mathbf{1}}s_g R_{(N_1 \to W)}\, {}^{\mathbf{1}}\vec{\beta}^{(N_1)}_{[k]} + {}^{\mathbf{1}}\vec{\beta}^{(W)}_{[1]} \tag{5.3}$$

$$ {}^{\mathbf{2}}\vec{\beta}^{(W)}_{[k]} = {}^{\mathbf{2}}s_g\, {}^{\mathbf{2}}\vec{\beta}^{(N_2)}_{[k]} \tag{5.4}$$

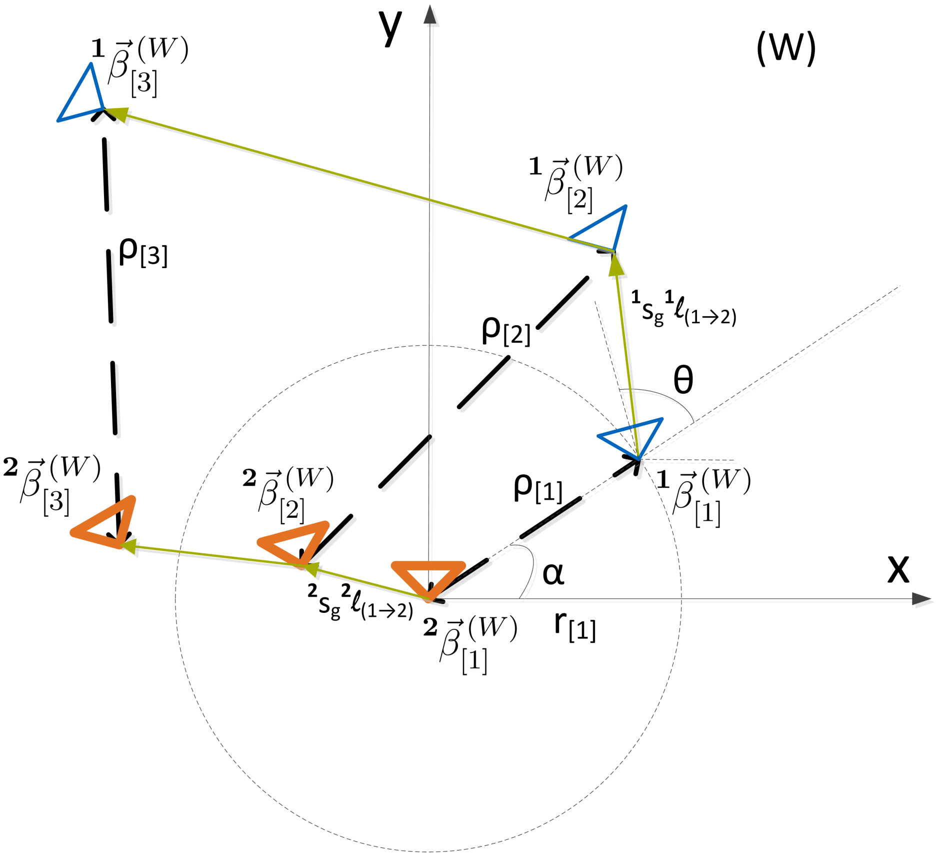

y
(W)
${}^{\mathbf{1}}\vec{\beta}^{(W)}_{[3]}$
${}^{\mathbf{1}}\vec{\beta}^{(W)}_{[2]}$
$\rho_{[3]}$
$\rho_{[2]}$
${}^{1}s_{g}\,{}^{1}\ell_{(1\rightarrow 2)}$
θ
$\rho_{[1]}$
${}^{\mathbf{2}}\vec{\beta}^{(W)}_{[3]}$
${}^{\mathbf{2}}\vec{\beta}^{(W)}_{[2]}$
${}^{\mathbf{1}}\vec{\beta}^{(W)}_{[1]}$
α
X
${}^{2}s_{g}\,{}^{2}\ell_{(1\rightarrow 2)}$
${}^{\mathbf{2}}\vec{\beta}^{(W)}_{[1]}$
$r_{[1]}$

Figure 5.1: Reference system and the geometry of the two rovers

Although the monocular camera itself can only estimate the motion with a scale ambiguity, with the additional help of a sparse set of noisy range measurements $\{\rho_{[k]}\}$, where

$$\rho_{[k]} = \left\| {}^{\mathbf{1}}\vec{\beta}^{(W)}_{[k]} - {}^{\mathbf{2}}\vec{\beta}^{(W)}_{[k]} \right\| + \eta_{[k]}, \tag{5.5}$$

a method for estimating the scale factors ${}^{\mathbf{1}}s_g, {}^{\mathbf{2}}s_g$ can be devised by exploiting consecutive ranging measurements at keyframes. The true range between the two rovers at time k is

$$\begin{aligned} G_k({}^{\mathbf{1}}s_g, {}^{\mathbf{2}}s_g, \alpha, \theta, r_{[1]}) = r_{[k]} = \left\| {}^{\mathbf{1}}\vec{\beta}^{(W)}_{[k]} - {}^{\mathbf{2}}\vec{\beta}^{(W)}_{[k]} \right\| \\ = \left\| {}^{\mathbf{1}}s_g R(\alpha + \theta - \frac{\pi}{2}) {}^{\mathbf{1}}\vec{\beta}^{(N_1)}_{[k]} + r_{[1]} R(\alpha)[1, 0]^T - {}^{\mathbf{2}}s_g {}^{\mathbf{2}}\vec{\beta}^{(N_2)}_{[k]} \right\|, \end{aligned} \tag{5.6}$$

which is determined by the rover trajectories in navigation frames and 5 unknown scalar parameters: the scale factors ${}^{\mathbf{1}}s_g, {}^{\mathbf{2}}s_g \in \mathbb{R}^+$,the polar angle $\alpha \in [0, 2\pi)$, the attitude angle $\theta \in [0, 2\pi)$, and the initial distance $r_{[1]} \in \mathbb{R}^+$. We stack them into a parameter vector $\xi = [{}^{\mathbf{1}}s_g, {}^{\mathbf{2}}s_g, \alpha, \theta, r_{[1]}]^T$.

Utilizing communication functionality of the radio link between the two rovers, rover 1 can transmit its estimated motion (up-to-scale) to rover 2. Rover 2 serves as the master that obtains both local trajectory estimates. Therefore, by using the available set of range measurements at time N_k, the unknown parameters can be estimated by minimizing

$$\hat{\xi} = \arg\min_{\xi} \|\rho - G(\xi)\|^2_{Q^{-1}}, \quad \text{s.t. } B\xi > 0, \tag{5.7}$$

where the vector $\rho = [\rho_{[1]}, \rho_{[2]}, ..., \rho_{[N_k]}]^T$ and $G(\xi) = [G_1(\xi), G_2(\xi), ..., G_{N_k}(\xi)]^T$.

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

is a selection matrix used to impose the positiveness of both scales and the initial distance. Q is the covariance matrix of the noise $\eta = [\eta_{[1]}, \eta_{[2]}, ..., \eta_{[N_k]}]^T$. The covariance of the range measurements can be obtained from the signal receiver. The Cramér-Rao lower bound of the range estimation can be used as an approximation when the covariance calculation is unavailable. If the ranging noise are uncorrelated, Q is a diagonal matrix.

Due to the bounded search space and the presence of several local minima, it is challenging to solve the nonlinear inequality constrained optimization in Eqn. (5.7). However, not all minima violating the constraints represent erroneous solution, due to the symmetric properties of the objective function. According to Eq. (5.6), the norm G_k is invariant if the vector ${}^{\mathbf{1}}\vec{\beta}^{(W)}_{[k]} - {}^{\mathbf{2}}\vec{\beta}^{(W)}_{[k]}$ is reversed in direction. Consequently, for any parameter vector ξ, the value of the object function is invariant to the following parameter change:

$$\begin{aligned} & G_k({}^{\mathbf{1}}s_g, {}^{\mathbf{2}}s_g, \alpha, \theta, r_{[1]}) \\ =& G_k(-{}^{\mathbf{1}}s_g, {}^{\mathbf{2}}s_g, \alpha, \theta + \pi, r_{[1]}) \\ =& G_k(-{}^{\mathbf{1}}s_g, -{}^{\mathbf{2}}s_g, \alpha + \pi, \theta + \pi, r_{[1]}) \\ =& G_k(-{}^{\mathbf{1}}s_g, -{}^{\mathbf{2}}s_g, \alpha, \theta, -r_{[1]}) \\ =& G_k(-{}^{\mathbf{1}}s_g, {}^{\mathbf{2}}s_g, \alpha + \pi, \theta, -r_{[1]}) \\ =& G_k({}^{\mathbf{1}}s_g, -{}^{\mathbf{2}}s_g, \alpha + \pi, \theta, r_{[1]}) \\ =& G_k({}^{\mathbf{1}}s_g, -{}^{\mathbf{2}}s_g, \alpha, \theta + \pi, -r_{[1]}) \\ =& G_k({}^{\mathbf{1}}s_g, {}^{\mathbf{2}}s_g, \alpha + \pi, \theta + \pi, -r_{[1]}). \end{aligned} \tag{5.8}$$

Table 5.1: Transformation on the results from unconstrained optimization.

If			Transformation				
${}^{\mathbf{1}}\hat{s}_g > 0$	${}^{\mathbf{2}}\hat{s}_g < 0$	$\hat{r}_{[1]} > 0$	${}^{\mathbf{1}}\hat{s}_g \leftarrow {}^{\mathbf{1}}\hat{s}_g$	${}^{\mathbf{2}}\hat{s}_g \leftarrow -{}^{\mathbf{2}}\hat{s}_g$	$\hat{\alpha} \leftarrow \hat{\alpha} + \pi$	$\hat{\theta} \leftarrow \hat{\theta}$	$\hat{r}_{[1]} \leftarrow \hat{r}_{[1]}$
${}^{\mathbf{1}}\hat{s}_g > 0$	${}^{\mathbf{2}}\hat{s}_g < 0$	$\hat{r}_{[1]} < 0$	${}^{\mathbf{1}}\hat{s}_g \leftarrow {}^{\mathbf{1}}\hat{s}_g$	${}^{\mathbf{2}}\hat{s}_g \leftarrow -{}^{\mathbf{2}}\hat{s}_g$	$\hat{\alpha} \leftarrow \hat{\alpha}$	$\hat{\theta} \leftarrow \hat{\theta} + \pi$	$\hat{r}_{[1]} \leftarrow -\hat{r}_{[1]}$
${}^{\mathbf{1}}\hat{s}_g > 0$	${}^{\mathbf{2}}\hat{s}_g > 0$	$\hat{r}_{[1]} < 0$	${}^{\mathbf{1}}\hat{s}_g \leftarrow {}^{\mathbf{1}}\hat{s}_g$	${}^{\mathbf{2}}\hat{s}_g \leftarrow {}^{\mathbf{2}}\hat{s}_g$	$\hat{\alpha} \leftarrow \hat{\alpha} + \pi$	$\hat{\theta} \leftarrow \hat{\theta} + \pi$	$\hat{r}_{[1]} \leftarrow -\hat{r}_{[1]}$
${}^{\mathbf{1}}\hat{s}_g < 0$	${}^{\mathbf{2}}\hat{s}_g > 0$	$\hat{r}_{[1]} > 0$	${}^{\mathbf{1}}\hat{s}_g \leftarrow -{}^{\mathbf{1}}\hat{s}_g$	${}^{\mathbf{2}}\hat{s}_g \leftarrow {}^{\mathbf{2}}\hat{s}_g$	$\hat{\alpha} \leftarrow \hat{\alpha}$	$\hat{\theta} \leftarrow \hat{\theta} + \pi$	$\hat{r}_{[1]} \leftarrow \hat{r}_{[1]}$
${}^{\mathbf{1}}\hat{s}_g < 0$	${}^{\mathbf{2}}\hat{s}_g < 0$	$\hat{r}_{[1]} > 0$	${}^{\mathbf{1}}\hat{s}_g \leftarrow -{}^{\mathbf{1}}\hat{s}_g$	${}^{\mathbf{2}}\hat{s}_g \leftarrow -{}^{\mathbf{2}}\hat{s}_g$	$\hat{\alpha} \leftarrow \hat{\alpha} + \pi$	$\hat{\theta} \leftarrow \hat{\theta} + \pi$	$\hat{r}_{[1]} \leftarrow \hat{r}_{[1]}$
${}^{\mathbf{1}}\hat{s}_g < 0$	${}^{\mathbf{2}}\hat{s}_g < 0$	$\hat{r}_{[1]} < 0$	${}^{\mathbf{1}}\hat{s}_g \leftarrow -{}^{\mathbf{1}}\hat{s}_g$	${}^{\mathbf{2}}\hat{s}_g \leftarrow -{}^{\mathbf{2}}\hat{s}_g$	$\hat{\alpha} \leftarrow \hat{\alpha}$	$\hat{\theta} \leftarrow \hat{\theta}$	$\hat{r}_{[1]} \leftarrow -\hat{r}_{[1]}$
${}^{\mathbf{1}}\hat{s}_g < 0$	${}^{\mathbf{2}}\hat{s}_g > 0$	$\hat{r}_{[1]} < 0$	${}^{\mathbf{1}}\hat{s}_g \leftarrow -{}^{\mathbf{1}}\hat{s}_g$	${}^{\mathbf{2}}\hat{s}_g \leftarrow {}^{\mathbf{2}}\hat{s}_g$	$\hat{\alpha} \leftarrow \hat{\alpha} + \pi$	$\hat{\theta} \leftarrow \hat{\theta}$	$\hat{r}_{[1]} \leftarrow -\hat{r}_{[1]}$

Therefore, due to the numerical symmetry property of the cost function, we can obtain the estimates of the parameters by solving the corresponding unconstrained problem and by transforming the results obtained with the relations given in Table 5.1.

For an unconstrained problem, the nonlinear optimization in Eqn. (5.7) can be solved by linearizing the measurement function at an initial point (the initialization method is discussed later), and by iteratively solving the linearized optimization and updating the estimated parameters as well as the linearization point as

$$\hat{\xi} = \arg\min_{\xi} \|\rho - J(\xi)\xi\|^2_{Q^{-1}} , \tag{5.9}$$

with Jacobian matrix

$$J(\xi) = \begin{bmatrix} \frac{\partial G_1(\xi)}{\partial {}^{\mathbf{1}}s_g} & \frac{\partial G_1(\xi)}{\partial {}^{\mathbf{2}}s_g} & \frac{\partial G_1(\xi)}{\partial \alpha} & \frac{\partial G_1(\xi)}{\partial \theta} & \frac{\partial G_1(\xi)}{\partial r_{[1]}} \\ \frac{\partial G_2(\xi)}{\partial {}^{\mathbf{1}}s_g} & \frac{\partial G_2(\xi)}{\partial {}^{\mathbf{2}}s_g} & \frac{\partial G_2(\xi)}{\partial \alpha} & \frac{\partial G_2(\xi)}{\partial \theta} & \frac{\partial G_2(\xi)}{\partial r_{[1]}} \\ \vdots & & & & \\ \frac{\partial G_{N_k}(\xi)}{\partial {}^{\mathbf{1}}s_g} & \frac{\partial G_{N_k}(\xi)}{\partial {}^{\mathbf{2}}s_g} & \frac{\partial G_{N_k}(\xi)}{\partial \alpha} & \frac{\partial G_{N_k}(\xi)}{\partial \theta} & \frac{\partial G_{N_k}(\xi)}{\partial r_{[1]}} \end{bmatrix} . \tag{5.10}$$

and the iterative update as

$$\hat{\xi}_{i+1} = \hat{\xi}_i + \left(J^T(\hat{\xi}_i)Q^{-1}J(\hat{\xi}_i)\right)^{-1} J^T(\hat{\xi}_i)Q^{-1}\left(\rho - G(\hat{\xi}_i)\right) . \tag{5.11}$$

If ${}^{\mathbf{1}}s_g$, ${}^{\mathbf{2}}s_g$ or $r_{[1]}$ are negative in the result, the parameters can be mapped to the symmetric solution in the valid search space as shown in Table 5.1. Consequently, the scales of the trajectories ${}^{\mathbf{1}}s_g$ and ${}^{\mathbf{2}}s_g$ are resolved. Additionally, the relative position and attitude between the two rovers are obtained. Combining with the trajectory estimates in navigation frames, the relative pose at any keyframe k can be estimated coarsely. As a distributed system, the master rover can transmit the estimation results to the other one using the communication system.

In order to solve the problem in Eqn. (5.9), theoretically $N_k \geq 5$ range measurements are required. Due to the high nonlinearity of the objective function, the Levenberg-Marquardt algorithm is applied, instead of a Gauss-Newton approach, in order to exploit its superior global minimization capabilities. In addition, the initialization of the optimization is crucial due to the presence of a number of local minima. Although a suboptimal solution may have similar residual as the global minimum, the estimated parameters can be far away from the true value, leading to a wrong scale or pose. While $\rho_{[1]}$ is a precise approximation of the initial range $r_{[1]}$ due to the high accuracy of ranging measurements, the scaling factor ${}^{\mathbf{1}}s_g$ and ${}^{\mathbf{2}}s_g$ are significantly insensitive to the global minima problem, provided that the selected keyframes are sufficiently spaced. The estimation of the polar angle α and the attitude angle θ presents larger difficulties. Fortunately, the parameters to be estimated are constants and in most cases they do not need to be updated at high frequency. Hence a serial search for the proper initialization of the two angles is feasible. It is remarkable

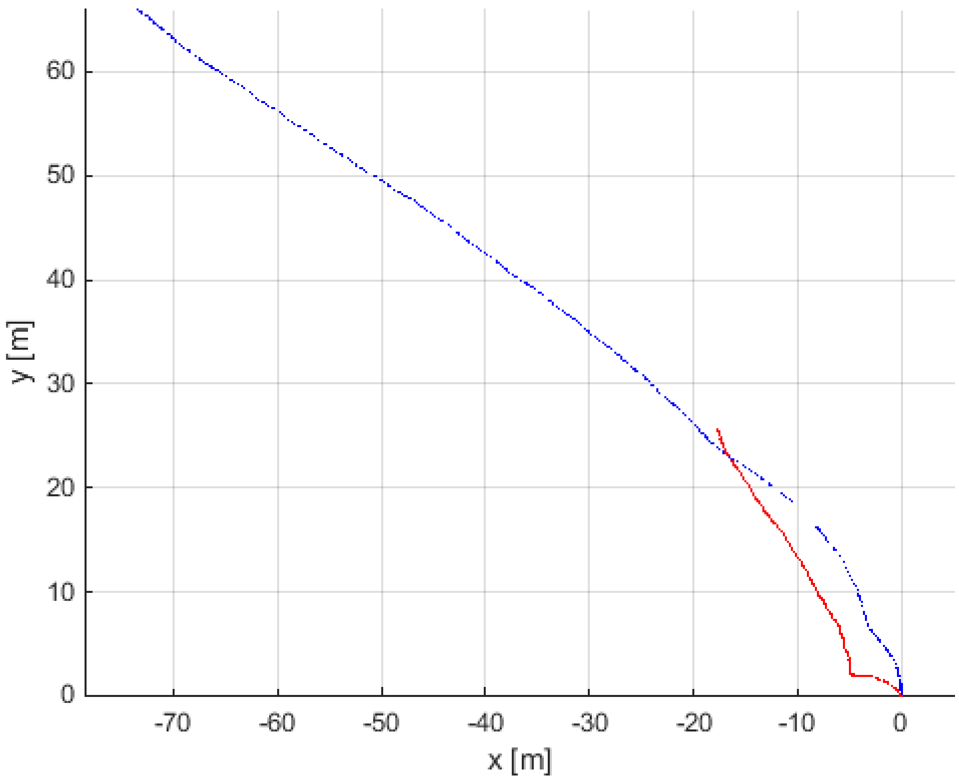


Figure 5.2: Trajectories of the two rovers in Scenario #1.

that if the relative position between the two rovers can be estimated by other methods, e.g., using ranging measurements from the swarm network in [5], the polar angle α could be precisely initialized. As a result, the search space would even reduce to a one-dimensional set. The requirement in initialization also explains why the problem with 3D motion is much more challenging. First of all, the local attitude angle have to be expressed with three orientation parameters and the initial relative position need to be parameterized by elevation and azimuth instead of α. Moreover, the objective function will have numerous local minima. As a result, with limited computational power, it is significantly challenging to obtain correct parameters in the 3D motion setup.

We test the proposed method on multiple trajectories using simulation data with Gaussian additive noise. The trajectories are generated with random walk processes as accelerations, starting from static locations with random relative position and attitude. In the simulation, there are two noise sources added on the measurements, the ranging noise with standard deviation σ_ρ and the translation noise with σ_t. In order to simulate a realistic scenario, the error of the trajectory estimation is added to all the translation estimates instead of on positions, i.e., the error accumulates over frames.

For the trajectories shown in Fig. 5.2 with 500 keyframes from each camera, the root-mean-square-error (RMSE) of the parameter estimation under different noise levels is shown in Table 5.2. All the RMSE are calculated with ten repetitive runs with independent noise. Rover 2, i.e., the master node, is plotted in blue and rover 1 is in red. Fig. 5.3 shows the first 30 frames of the rovers to illustrate the initial relative geometry more clearly. In the serial search of initial values of the polar angle α and attitude angle θ, the grid size is set to be 10 degrees in the simulation. It can be concluded from the results that in this scenario, the optimization converges well even for 5 [cm] translation error and 20 [cm] ranging noise. The scale factor of the trajectories for both rovers can be accurately estimated. An improvement in estimation precision

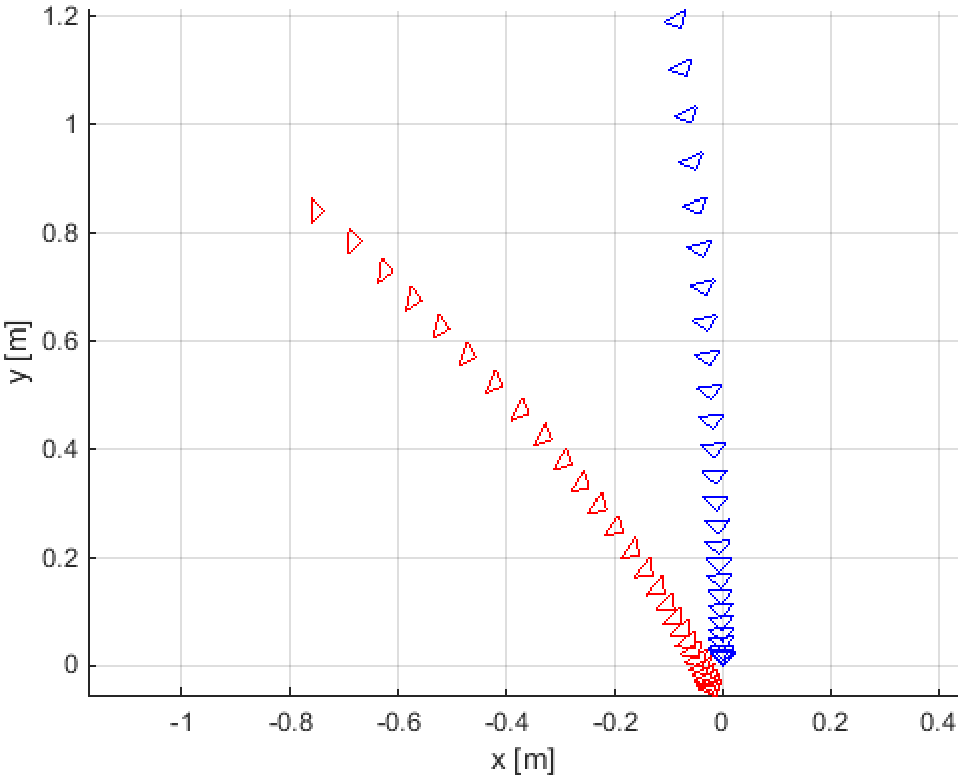


Figure 5.3: First 30 frames of the two rovers in Scenario #1.

Table 5.2: Estimation error of scales and pose parameters in Scenario #1.

σ_t	1 [cm]	1 [cm]	3 [cm]	5 [cm]	5 [cm]
σ_ρ	1 [cm]	10 [cm]	10 [cm]	10 [cm]	20 [cm]
$RMSE(^{\mathbf{1}}s_g)$	0.0016	0.0049	0.0120	0.0129	0.0086
$RMSE(^{\mathbf{2}}s_g)$	0.0015	0.0045	0.0127	0.0116	0.0067
$RMSE(\alpha)$ [deg]	3.3893	3.3426	8.2128	6.9004	8.8601
$RMSE(\theta)$ [deg]	0.6539	1.8069	4.2246	8.5225	6.2905
$RMSE(r_{[1]})$ [m]	0.0171	0.0301	0.0596	0.1620	0.0716

of the angles can be obtained by setting a higher density of serial search values in the initialization of the non-linear optimization.

Other scenarios are also simulated to test the performance of the proposed method in different motion geometries. The trajectories of the rovers in various scenarios are shown in Fig. 5.4 and the corresponding estimation results are given in Table 5.3. It can be concluded that the method performs well in various scenarios with different geometries. A key factor that affects the precision of the estimation is the magnitude of the simulated motion. If the change of distance between the two rovers is comparable to the ranging noise, the measurement noise would be dominant in the estimation.

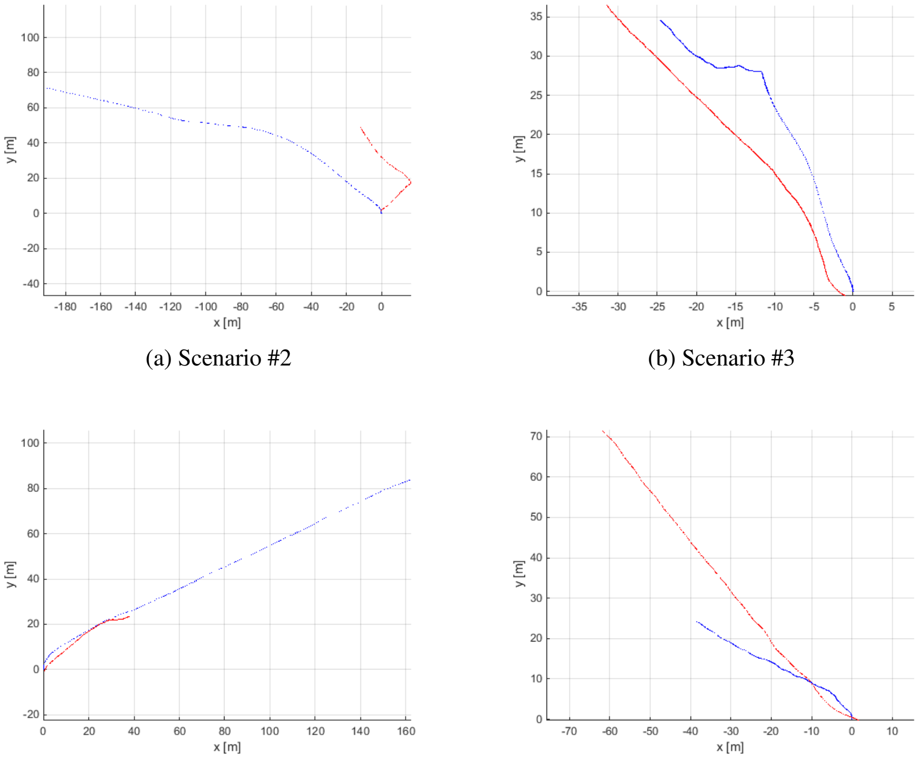
5.2 Cooperative Visual SLAM with a Ranging Link

5.2.1 Tight Coupling of Cameras and Ranging Measurements for a Pair of Rovers

As shown in the previous section, by utilizing range measurements between two dynamic rovers, the global scale of monocular cameras on both rovers can be obtained from Eqn. (5.7), and the relative pose parameters

Table 5.3: Estimation error of parameters in various scenarios.

Scenario #	2	3	4	5
σ_t	5 [cm]	5 [cm]	3 [cm]	1 [cm]
σ_ρ	20 [cm]	10 [cm]	10 [cm]	5 [cm]
$RMSE({}^{\mathbf{1}}s_g)$	0.0052	0.0030	0.0149	0.0009
$RMSE({}^{\mathbf{2}}s_g)$	0.0018	0.0017	0.0048	0.0033
$RMSE(\alpha)$ [deg]	2.9416	9.7634	9.0721	5.8405
$RMSE(\theta)$ [deg]	3.5057	6.0490	8.6343	0.7336
$RMSE(r_{[1]})$ [m]	0.1337	0.0717	0.0486	0.0205



(a) Scenario #2

(b) Scenario #3

(c) Scenario #4

(d) Scenario #5

Figure 5.4: Trajectories of the two rovers in various scenarios.

between the two trajectories can be estimated as $[\hat{\alpha}, \hat{\theta}, \hat{r}_{[1]}]$. Following the denotation in Eqn. (2.42), the coarse estimates of the two vehicles' pose in the global reference frame (W) can be estimated accordingly as

$$ {}^{\mathbf{1}}\hat{x}^{(W)}_{[k]} = \begin{bmatrix} {}^{\mathbf{1}}\hat{\beta}^{(W)}_{[k]} \\ {}^{\mathbf{1}}\hat{\phi}^{(W)}_{[k]} \end{bmatrix}, \quad {}^{\mathbf{2}}\hat{x}^{(W)}_{[k]} = \begin{bmatrix} {}^{\mathbf{2}}\hat{\beta}^{(W)}_{[k]} \\ {}^{\mathbf{2}}\hat{\phi}^{(W)}_{[k]} \end{bmatrix}, \tag{5.12} $$

with

$$ {}^{\mathbf{1}}\hat{\beta}^{(W)}_{[k]} = {}^{\mathbf{1}}\hat{s}_g R(\hat{\alpha} + \hat{\theta} - \frac{\pi}{2})\,{}^{\mathbf{1}}\vec{\beta}^{(N_1)}_{[k]} + \hat{r}_{[1]} R(\hat{\alpha})[1,0]^T, \quad {}^{\mathbf{2}}\hat{\beta}^{(W)}_{[k]} = {}^{\mathbf{2}}\hat{s}_g\, {}^{\mathbf{2}}\vec{\beta}^{(N_2)}_{[k]}, \tag{5.13} $$

and

$$ {}^{\mathbf{1}}\hat{\phi}^{(W)}_{[k]} = A\left(\hat{R}_{(N_1 \to W)} \prod_{i=2}^{k} \hat{R}_{({}^{j}i \to {}^{j}i-1)}\right), \quad {}^{\mathbf{2}}\hat{\phi}^{(W)}_{[k]} = A\left(\prod_{i=2}^{k} \hat{R}_{({}^{j}i \to {}^{j}i-1)}\right), \tag{5.14} $$

where $A(\cdot) : \mathbf{SO}(3) \to \mathbb{R}$ is the function that extract heading angle ϕ from the corresponding rotation matrix, and the relative attitude change can be obtained from the visual navigation algorithm.

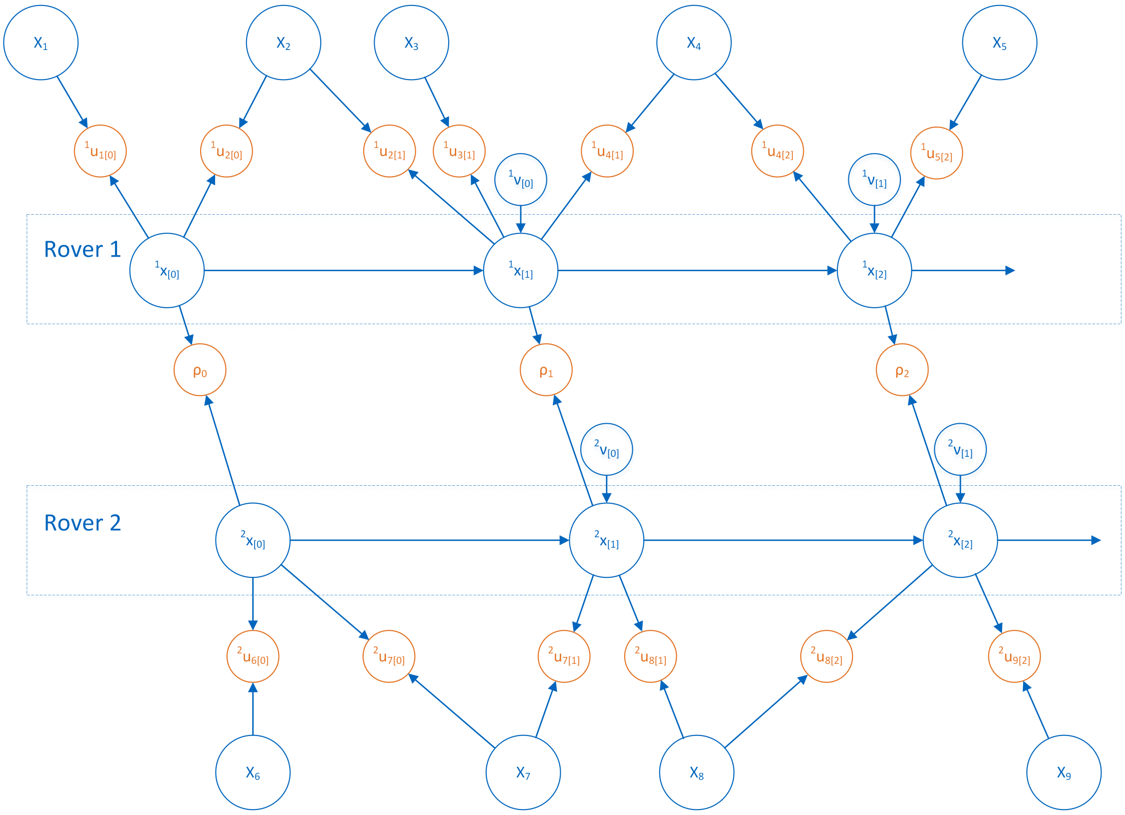


Figure 5.5: Bayesian network of the relative pose estimation of two rovers.

In order to refine the result, by integrating the 2D feature location and ranging measurements, the trajectory of both rovers can be estimated in a tightly coupled sensor fusion method. Similarly as in Section 4.4, the error in scales ${}^{\mathbf{1}}\hat{s}_g$, ${}^{\mathbf{2}}\hat{s}_g$ and in monocular egomotion ${}^{\mathbf{1}}\hat{\beta}^{(N)}_{[k]}$, ${}^{\mathbf{2}}\hat{\beta}^{(N)}_{[k]}$ do not need to be considered separately. Instead, a set of positions in global frame $\{{}^{\mathbf{1}}\vec{\beta}^{(W)}_{[k]}, {}^{\mathbf{2}}\vec{\beta}^{(W)}_{[k]} | k = 1, ..., N_k\}$ is refined by combining the estimated scale and positions in navigation frame as initial coarse estimates using Eqn. (5.13).

Fig. 5.5 shows the Bayesian network of a tight coupling sensor fusion method exploiting both the visual measurements and the ranging measurements. Applying the dependency among the random variables in the Bayesian network, the poses of the rover pair can be obtained from the sensor fusion by the following maximum likelihood estimator:

$$\begin{aligned}&\left\{{}^{\mathbf{1}}\hat{x}^{(W)}_{[k]}, {}^{\mathbf{2}}\hat{x}^{(W)}_{[k]}, \vec{X}^{(W)}_i\right\}\\ &= \arg\max \prod_{k=1}^{N_k}\prod_{i=1}^{N_p} p({}^{\mathbf{1}}\mu_{i[k]}|\pi(\vec{X}^{(W)}_i, {}^{\mathbf{1}}x_{[k]})^{v_{1,i[k]}} p({}^{\mathbf{2}}\mu_{i[k]}|\pi(\vec{X}^{(W)}_i, {}^{\mathbf{2}}x_{[k]})^{v_{2,i[k]}} p(\rho_{[k]}|{}^{\mathbf{1}}x_{[k]}, {}^{\mathbf{2}}x_{[k]}),\end{aligned} \tag{5.15}$$

where $v_{1,i[k]}$ and $v_{2,i[k]}$ are the visibility mask for the two cameras respectively.

Under Gaussian noise assumption, the maximum likelihood estimation is equivalent to the following least squares problem:

$$\begin{aligned}&\left\{{}^{\mathbf{1}}\hat{x}^{(W)}_{[k]}, {}^{\mathbf{2}}\hat{x}^{(W)}_{[k]}, \vec{X}^{(W)}_i\right\}\\ &= \arg\min \sum_{k=1}^{N_k} \chi_k({}^{\mathbf{1}}x^{(W)}_{[k]}, {}^{\mathbf{2}}x^{(W)}_{[k]}) + \sum_{k=1}^{N_k}\sum_{i=1}^{N_p}(S_{ik}({}^{\mathbf{1}}x^{(W)}_{[k]}, \vec{X}^{(W)}_i) + S_{ik}({}^{\mathbf{2}}x^{(W)}_{[k]}, \vec{X}^{(W)}_i)).\end{aligned} \tag{5.16}$$

In the fusion, $S_{ik}(\cdot) = v_{i[k]} \left\| \mu_{i[k]} - \pi(\vec{X}_i, x_{[k]}) \right\|^2_{\Sigma^{-1}_{u,ik}}$, and $\chi_k(\cdot)$ is defined as

$$\chi_k = w_k \left(\left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^{\mathbf{1}}x^{(W)}_{[k]} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^{\mathbf{2}}x^{(W)}_{[k]} \right\| - \rho_{[k]} \right)^2 , \tag{5.17}$$

where $w_k = (E\{\eta^2_{[k]}\})^{-1}$. Due to the nonlinear nature of the problem, the coarse estimates are important as proper initial values to ensure the convergence of the optimization. Using the coarse estimates as initial value, the optimal estimation of the rovers' poses is obtained by solving the non-linear optimization in Eqn. (5.16). An iterative numerical solver such as Levenberg-Marquardt algorithm [111] can be used to solve the optimization. Due to the sparsity of the measurement space, the optimization can also be carried out in an incremental way by applying probability inference techniques such as [49].

The fusion algorithm does not require any common field-of-view for the two cameras, making the proposed approach more flexible and efficient in exploration tasks. The two navigation tasks are only coupled by the ranging, and remain otherwise independent.

5.2.2 CRLB of Cooperative Visual SLAM

Using the uncertainty model introduced in Section 3.3, the positioning accuracy using cameras is indicated by the corresponding Cramér-Rao lower bound (CRLB). Similarly, the performance of the tightly coupled sensor fusion using camera and ranging proposed in the previous section can also be reflected by the CRLB.

Stack all the measurements $\{{}^{\mathbf{1}}\mu_{i[k]}\}$, $\{{}^{\mathbf{2}}\mu_{i[k]}\}$ and $\{\rho_{[k]}\}$ into a vector λ, and all the parameters $\{{}^{\mathbf{1}}x^{(W)}_{[k]}\}$, $\{{}^{\mathbf{2}}x^{(W)}_{[k]}\}$, and $\{\vec{X}^{(W)}_i\}$ into $\Theta \in \mathbb{R}^{3N_p+6N_k}$, the CRLB of the estimated parameters is

$$CRLB(\Theta) = I^{-1}_{\Theta} = -\left(E\left\{\nabla^2 \log\left(p(\lambda|\Theta)\right)\right\}\right)^{-1}. \tag{5.18}$$

The log-likelihood function $\log(p(\lambda|\Theta))$ can be calculated using $F_{ik}(\cdot)$ and $\chi_k(\cdot)$ in Eqn. (5.16).

A representative trajectory, shown in Fig. 5.6, is generated to evaluate the CRLB in a simulation scenario. We set 10000 feature points which are distributed randomly in the 3D space. The cameras' intrinsic parameters and sensor model are provided by a real camera with a resolution of 1024*768 pixels and pixel density $\approx$ 213.33 [pixels/mm]. The 2D features are generated by using perspective projection with visibility check. Gaussian white noise is added on both the 2D feature locations and the simulated range measurements.

Fig. 5.7 shows the CRLB with respect to the ranging accuracy, represented by standard deviation of the ranging noise. The y-axis is the CRLB for the x-component of rover 2's position. In the plot, the feature measurement noise level is $\sigma_u = 0.1$ [pixel]. It can be shown from the plot that when the ranging noise is small, the CRLB of the fusion-based method is much lower than the vision-only approach. When the ranging noise level is high, the fusion algorithm's accuracy converge to the vision-only method.

Fig. 5.8 illustrates the relation between the CRLB and the feature location accuracy. In this scenario, the ranging accuracy is fixed to 0.5 [m]. The performance of the vision-only approach degrades significantly for feature location noise with standard deviation greater than 1 pixel. On the other hand, the bound for the fusion algorithm is much lower with the aid of the ranging measurements. Similar results can also be concluded for other components of the estimated parameters.

5.2.3 Simulation Results of Cooperative Visual SLAM

The trajectories of two rovers, shown in Fig. 5.9, are generated to evaluate the proposed cooperative sensor fusion method in a simulated scenario. The left trajectory is from rover 1, while the right one is from rover 2. We set 10000 feature points distributed randomly in the 3D space. The intrinsic parameters and sensor model are those of a real camera. The image sensor has a resolution of 1024*768 pixels, with pixel density

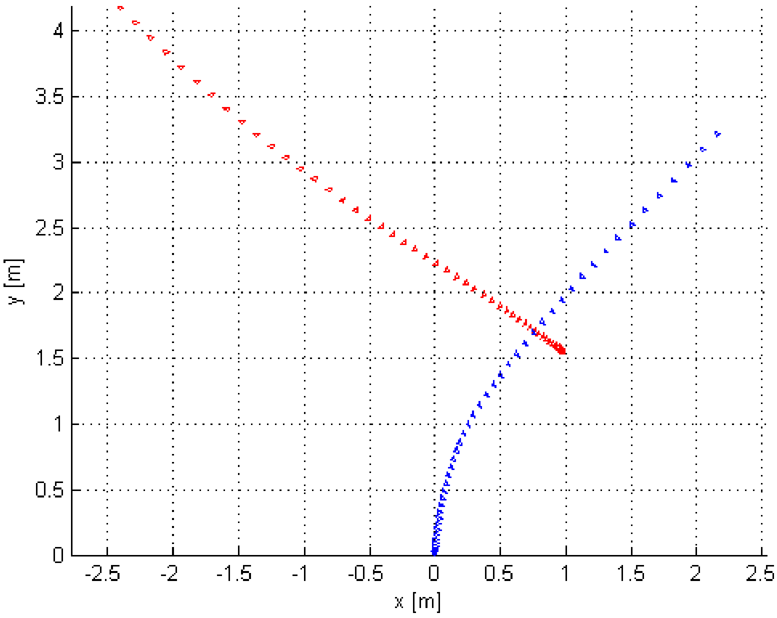


Figure 5.6: First 50 keyframes of the trajectory

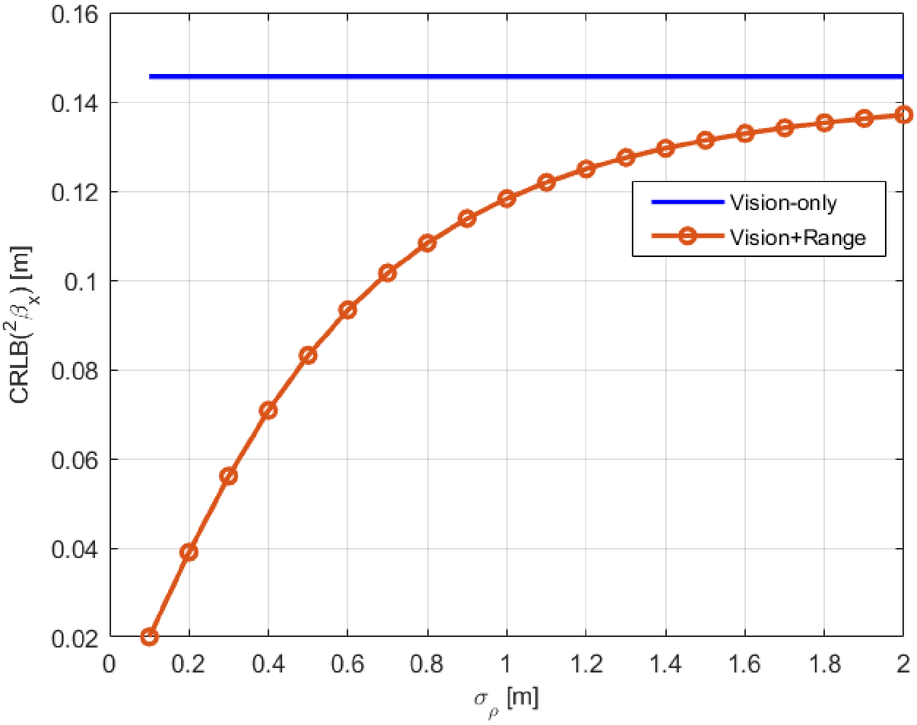


Figure 5.7: Change of CRLB with respect to σ_ρ, $\sigma_u = 0.1$ [pixel]

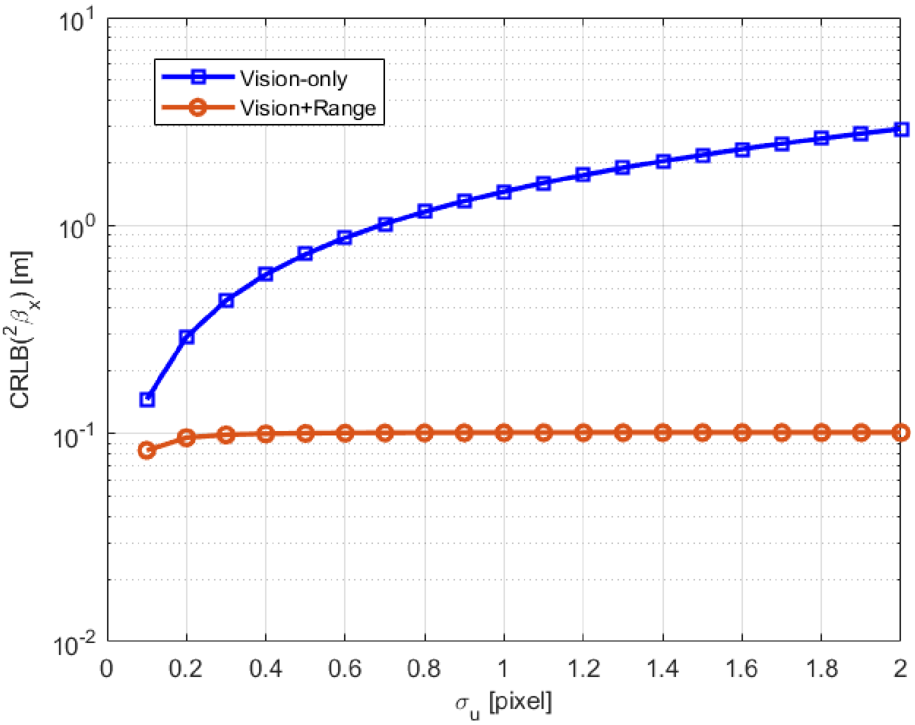


Figure 5.8: Change of CRLB with respect to σ_u, $\sigma_\rho = 0.5$ [m]

$\approx$ 213.33 [pixels/mm]. The focal length of the lenses is 2.5 [mm]. The 2D features are generated by using perspective projection with visibility check. White noise is added on both the 2D feature locations and the simulated range measurements.

The egomotion of the cameras is estimated coarsely using frame-by-frame visual odometry. To improve the accuracy, the rover poses and map point locations are refined using global optimization, either with VSLAM-only approach, i.e., bundle adjustment, or with the proposed sensor fusion approach exploiting the ranging measurements between the two rovers. The true global scale is applied for the vision-only approach for better comparison, which is unavailable in practice. The performance of the methods are shown in Fig. 5.10 and Fig. 5.11. In these two plots, the uncertainty of the feature location is 1 pixel, and the standard deviation of the ranging noise is 0.9 [m]. The two figures shows the trajectory of rover 1. Fig. 5.10 is a zoomed-in plot for a few representative keyframes in the trajectory. The red triangles denote the ground truth of the camera poses. The magenta poses are the outcomes of the visual odometry, which are used as the initial values in the optimization. Due to the error accumulation, the magenta trajectory drifts gradually away from the true one. The green trajectory shows the estimation result of the camera-only bundle adjustment, while the blue one shows the sensor fusion outcome when using both visual and ranging measurements.

It can be seen from the plots that the drifts in visual odometry can be mitigated by both global optimization methods, but the sensor fusion algorithm outperforms the vision-only approach in accuracy. Similar conclusions can be drawn for larger ranging noise. Fig. 5.12 and Fig. 5.13 illustrate the estimated trajectories from both approaches with $\sigma_\rho = 1.7[m]$. The accuracy of the sensor fusion method is still slightly better.

Fig. 5.14 plots the root mean square error (RMSE) of the camera poses as function of the change of the ranging noise level. Since the latter does not affect the VSLAM algorithm, the error of the bundle adjustment approach remains mostly unchanged.

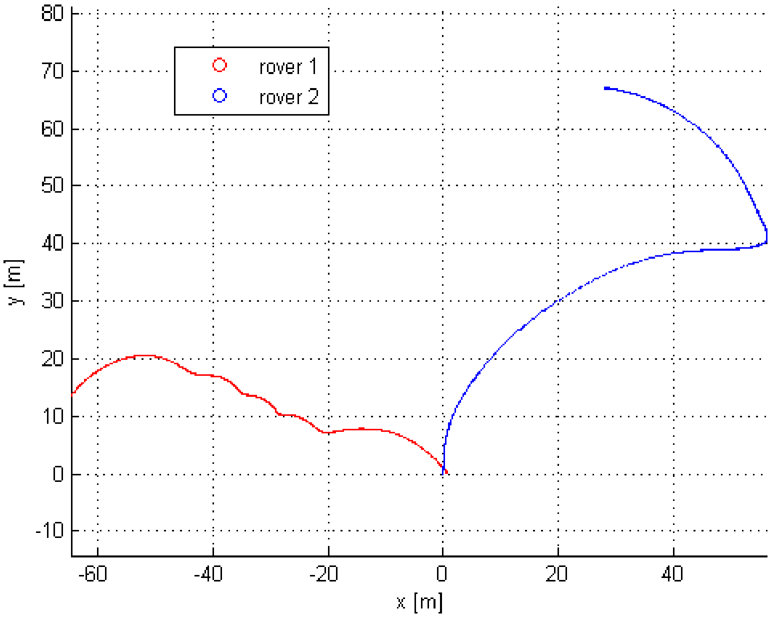


Figure 5.9: The trajectories of the two rovers

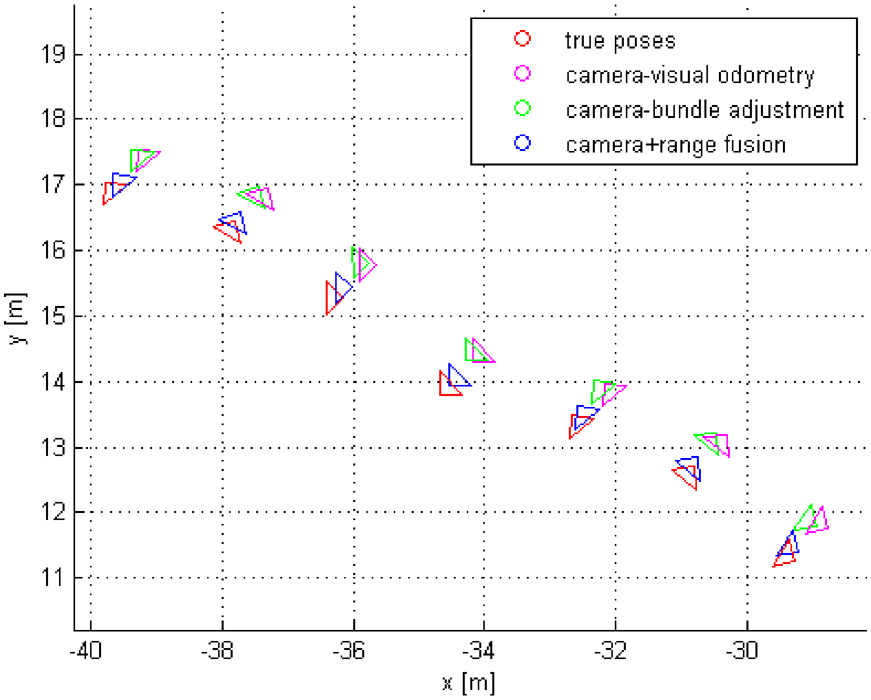


Figure 5.10: A segment of the trajectory of rover 1 estimated using different methods, $\sigma_\rho = 0.9[m]$

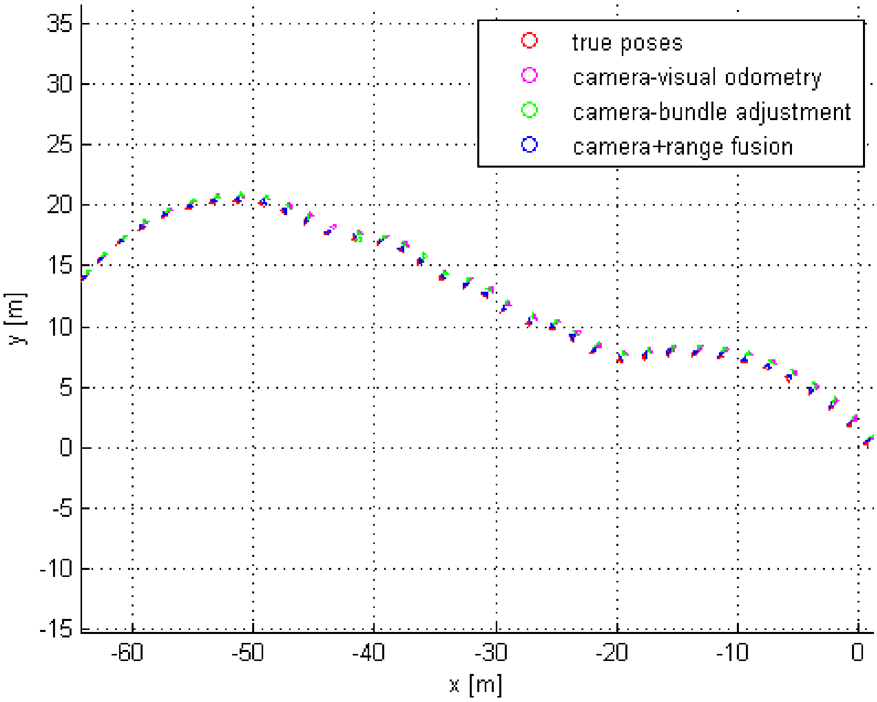


Figure 5.11: The trajectory of rover 1 estimated using different methods, $\sigma_\rho = 0.9[m]$

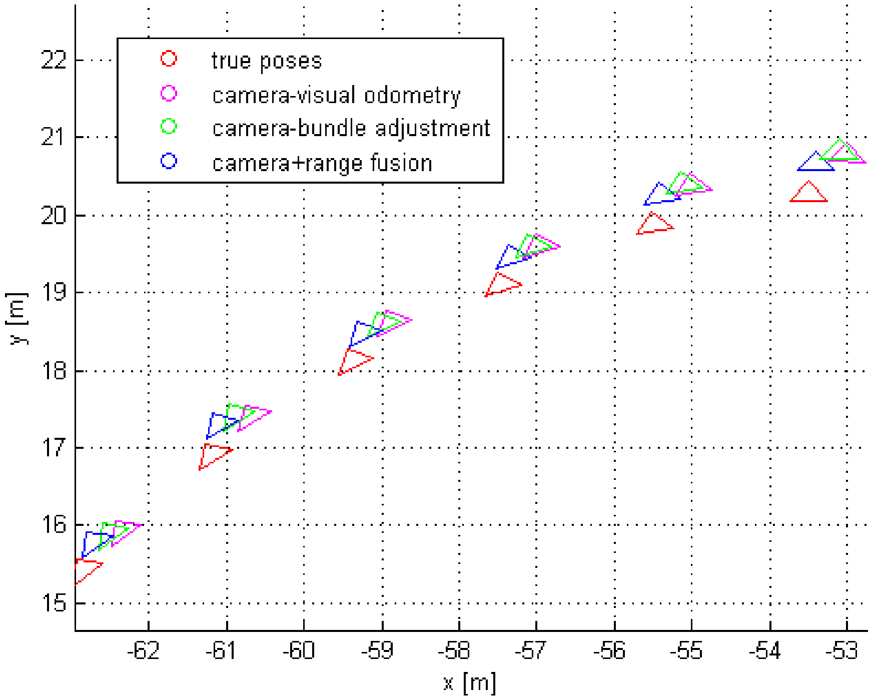


Figure 5.12: The zoomed in trajectory of rover 1 estimated using different methods, $\sigma_\rho = 1.7[m]$

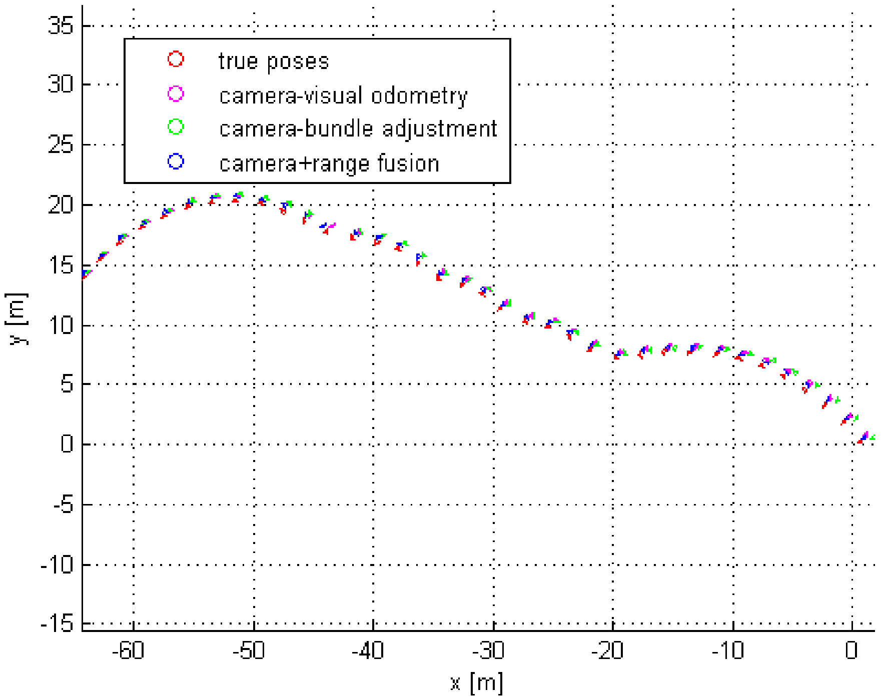


Figure 5.13: The trajectory of rover 1 estimated using different methods, $\sigma_\rho = 1.7[m]$

In conclusion, the sensor fusion approach significantly outperforms the vision-only method when the ranging noise is low. The performance of the proposed fusion method reduces to the one of classic VSLAM when the ranging measurement noise becomes very large (above meter level).

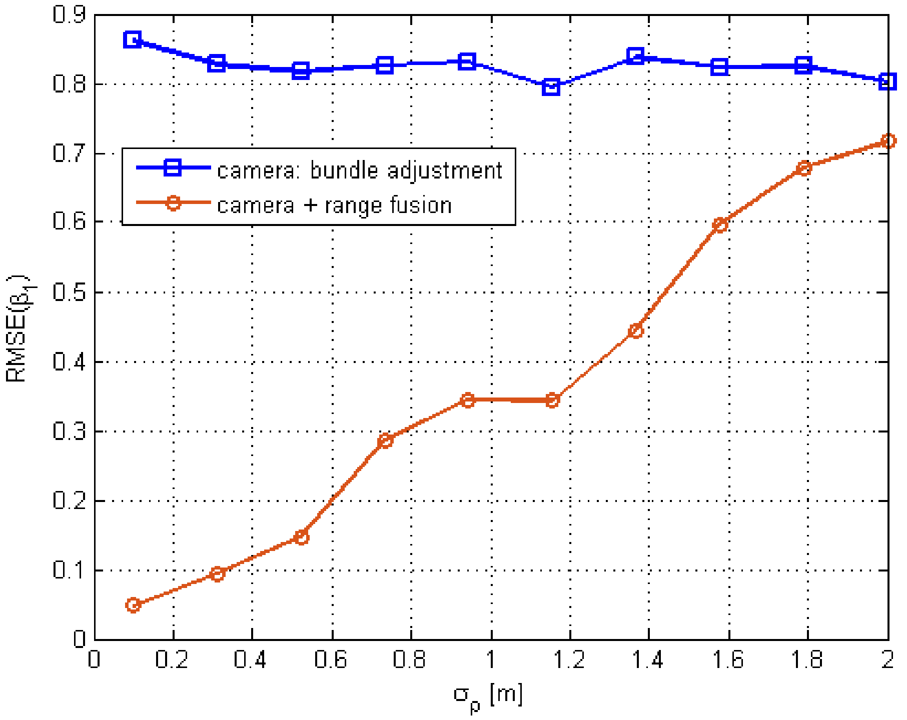


Figure 5.14: The RMSE of the rover poses with respect to the ranging noise level

5.3 Common Field-of-View Detection of a Vehicle Pair [1]

After knowing the relative pose between two rovers, the two monocular cameras on the rovers can establish a stereo vision system with a variable baseline. If the communication link between the two rovers supports sufficient data transmission capacity, one of the rover can transmit important images to the other for joint scene reconstruction. Exploiting the stereo vision, the common visible parts of the images can be reconstructed using state-of-the-art stereo 3D reconstruction algorithms such as the semi-global matching (SGM) proposed by Hirschmüller [112]. Compared with the structure from motion approach using a single rover, the variable baseline stereo in a robotic swarm has significant advantage in exploration efficiency, because the rovers do not have to move back and forth to execute the 3D reconstruction, especially for long baseline cases.

However, the automatic detection of overlapping regions in images of a scene taken by several cameras is difficult, even if the relative positions and orientations of the cameras are known. The overlapping region is both a function of the angle and distance of the object to the image plane, as illustrated in Fig. 5.17.

Local feature extraction is essential in feature based VSLAM algorithm, and the feature points can also be exploited to detect the common field-of-view (FOV). Conventional FOV detection methods, e.g., Snavely's approach in [113], are usually based on counting matching feature points. Bruckner et al. [114] improved the approach by releasing the constraints on the correctness of the feature matching so that it is more robust to outliers and matching errors. The approaches of extracting various sorts of local features are reviewed in [115]. Nevertheless, the image regions not detected as features can also belong to the overlapping area, which should be recognizable as common FOV regions as well. Fig. 5.15 and Fig. 5.16 shows an instance that more matched features does not necessarily result in larger common FOV of the two images.

[1] This part of work was joint work with Christoph Bamann, Technische Universität München

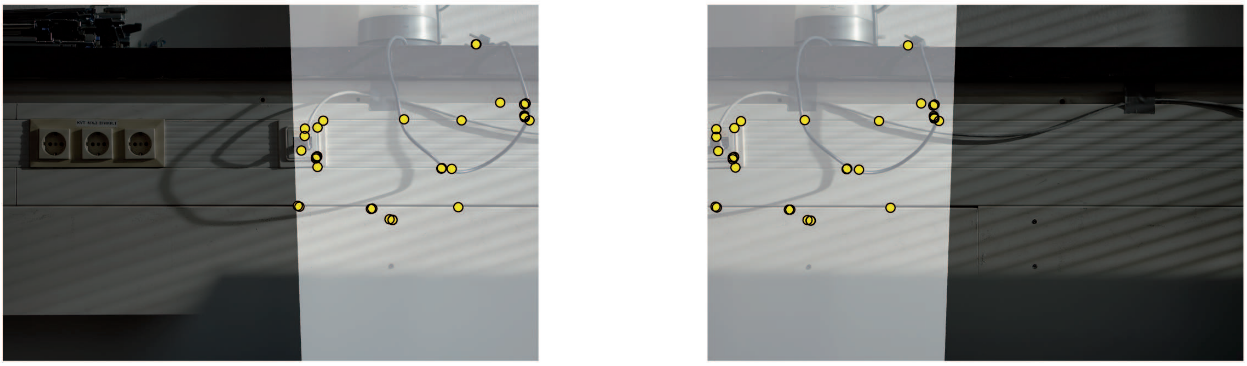


Figure 5.15: Small common FOV with many features.

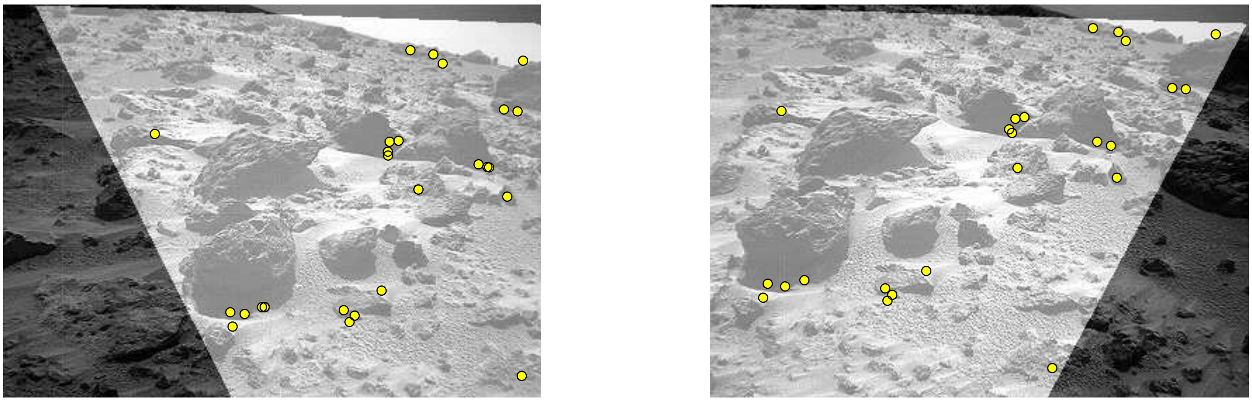


Figure 5.16: Large common FOV with not many features

To realize the goal of detecting overlapping regions, image segmentation can be a solution. However, the computational complexity is too high for realtime processing. It is sufficient to localize and track the feature points in our mission, but not essential to completely understand the content of the image. Therefore, in order to detect the common FOV efficiently, we propose a plane detection based approach that can fulfill the target of common FOV detection in various scenarios while without using the complex segmentation algorithm.

In many environments, the assumption that most feature points are approximately distributed on nearly planar surfaces or facets is valid. In some environments, the majority of features is close to a plane, e.g. the ground plane in the exploration of foreign planets. The statement is also valid for the facet planes in structured areas. Also, for cameras with finite resolution, the features far away can also be treated as laying on an infinite plane. This property is used in navigation, as in work of Xiao et al. [116] and Zhou et al. [117], and provides reasonable results, if a large number of feature points can be detected. The method proposed in this section does not require the features to be on a plane anymore - instead they can be approximately on a multitude of planes that are automatically determined, and with outliers might even be far away from those planes.

Without loss of generality, we consider the stereo vision with two cameras which are mounted on two distinct robots in a swarm with time-varying baseline. With known initialization, e.g., using the method proposed in Section 5.1 and Section 5.2, the relative pose between both cameras can be estimated, and the pose change for each camera can be locally tracked using VSLAM and shared over the communication channel efficiently. Consequently, it is acceptable to assume that the relative orientation as well as the baseline of the two cameras are known. This allows to triangulate the matched feature points from both views to obtain 3D coordinates. Provided the set of 3D points, we use fussy clustering algorithm to cluster the feature points to a few planes. The number of the planes can be calculated adaptively. The features

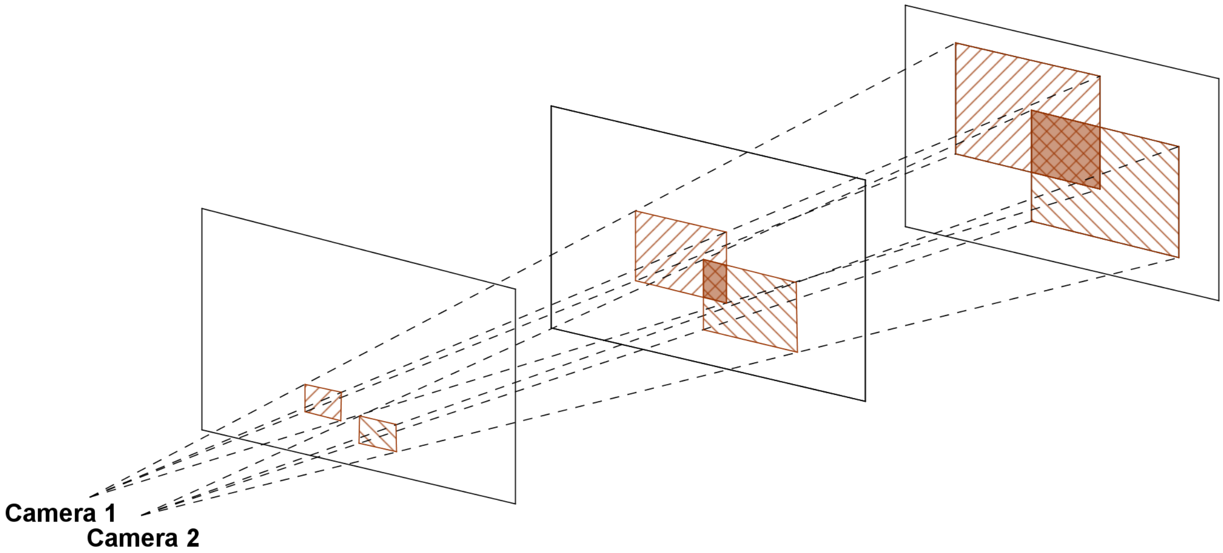


Figure 5.17: Depth Impact on Common Field-of-View

that are far away from any major plane are clustered as an outlier class, which will not be used in the following common FOV detection step in Section 5.3.2, in order to increase the robustness of the method. The applied adaptive fussy plane clustering method is introduced in Section 5.3.1. Moreover, the common FOV detection method proposed in this work is invariant to baseline scale change between the two cameras. As a result, the accuracy of the baseline length measurement does not affect the performance of the common FOV algorithm, as shall be seen in Section 5.3.3.

In the following subsections, we use both Cartesian coordinates describing points in Euclidian space $\mathbb{R}$ and homogeneous coordinates describing points in projective space $\mathbb{P}$. The latter have an additional scale coordinate compared with the Cartesian coordinates with same space dimension. In addition, the L2-norm (Euclidean distance) of a vector is denoted by $\|\cdot\|_2$, and $\langle\cdot,\cdot\rangle$ denotes the inner product of two vectors.

5.3.1 Adaptive Fussy Plane Clustering

Assume that N_p matched feature points with triangulated 3D coordinates $\vec{X}_1, \vec{X}_2, ..., \vec{X}_{N_p} \in \mathbb{R}^3$ are the only geometric information about the environment. Then hard clustering defines an initial allocation of the 3D points to M clusters by minimizing the overall distances between data points and centers of the clusters. The cluster number M should be initialized with a relative large number, and the similar clusters will be merged afterwards, as introduced later in this Section. This allocates every point to a unique cluster. Alternatively, fuzzy clustering defines a level of membership $w_{im} \in [0,1]$ of point i in cluster m [118]. The center $\vec{O}_m \in \mathbb{R}^3$ of cluster m is defined by

$$\vec{O}_m = \frac{\sum_{i=1}^{N_p}(w_{im})^{\vartheta}\vec{X}_i}{\sum_{i=1}^{N_p}(w_{im})^{\vartheta}}. \tag{5.19}$$

and the weights are obtained by minimizing the weighted squared distance from those centers:

$$\sum_{i=1}^{N_p}\sum_{m=1}^{M}(w_{im})^{\vartheta}\|\vec{X}_i - \vec{O}_m\|_2^2. \tag{5.20}$$

The exponent ϑ is called fuzzifier. It characterizes the extent of overlap among different clusters. When $\vartheta \leq 1$, the optimization becomes equivalent to hard clustering. The cluster centers $\vec{O}_m$ and membership levels w_{im} can be calculated iteratively as in the widely used algorithm proposed in [118]. This defines the initialization of the algorithm.

The points in the individual clusters should be in a plane. In order to determine that, one defines the cluster covariance matrix for the m-th cluster by

$$F_m = \frac{\sum_{i=1}^{N_p} (\vec{X}_i - \vec{O}_m)(w_{im})^{\vartheta}(\vec{X}_i - \vec{O}_m)^T}{\sum_{i=1}^{N_p} (w_{im})^{\vartheta}}. \tag{5.21}$$

The eigenvalues $\lambda_{1m} \leq \lambda_{2m} \leq \lambda_{3m}$ of F_m determine the shape of the cluster prototype. In the present context, the prototypes should be planes, i.e. $\lambda_{1m} \sim 0$. In the case that $\lambda_{1m}/\lambda_{2m}$ is larger than a threshold, the cluster is disregarded.

The allocation of points to clusters was rather arbitrary so far. In the present paper, a large number of cluster prototypes is used. The task is thus to merge clusters that are associated with the same plane. In a first step, we define a metric for measuring the distance from the planes associated with cluster m. Let e_{1m} be the normalized first eigenvector of F_m, then this measure is

$$h_{im} = |\langle e_{1m}, \vec{X}_i - \vec{O}_m \rangle|^2. \tag{5.22}$$

In order to handle noise and outliers, we introduce a 0-cluster. With these definition the weights are recomputed by minimizing:

$$\sum_{i=1}^{N_p} \sum_{m=1}^{M} (w_{im})^{\vartheta} \|h_{im}\|_2^2 + \sum_{i=1}^{N_p} (w_{io})^{\vartheta} \delta^2. \tag{5.23}$$

The rightmost summand makes the clustering process robust. If the distances of point $\vec{X}_i$ to all the other cluster prototypes are large compared to δ, its membership level w_{io} to the outlier cluster is high, so that it does not affect the calculation of the inlier prototypes.

Next clusters are compared: two clusters m_1 and m_2 are merged whenever the fuzzy inclusion similarity measure $\chi_{m_1 m_2}$ between them exceeds a threshold, which is normally chosen to be $1/(M-1)$ [119], i.e. when

$$\chi_{m_1 m_2} = \frac{\sum_{i=1}^{N_p} \min(w_{im_1}, w_{im_2})}{\min(\sum_{i=1}^{N_p} w_{im_1}, \sum_{i=1}^{N_p} w_{im_2})} > \frac{1}{M-1}. \tag{5.24}$$

The set of new clusters leads to a new computation of the centers $\vec{O}_m$ according to Equation (5.19) and of the cluster covariance matrix F_m according to Equation (5.21) and to another iteration of the algorithms. The iteration ends, after the first iteration without merging.

The eigenvectors e_{2m} and e_{3m} describe the plane γ_m associated with the m-th cluster after merging. This is the basis for computing the common field of view. Let M' be number of such planes.

5.3.2 Common Field-of-View Detection

The field of view of a camera is obtained by computing the first intersection of rays with the M' planes. Thus projecting the visible region of one camera on the M' planes and then on the image plane of another camera yields their common FOV.

The visible region of Camera 1 on the detected planes is defined both by the intersections of the planes and the crop due to the limits of the camera's sensor size. According to the projective geometry as introduced in [20], each plane i can be expressed by homogeneous coordinates $\tilde{\gamma}_i \in \mathbb{P}^3$, so that any points $\tilde{X} \in \mathbb{P}^3$ in the plane satisfies $\tilde{\gamma}_i^T \tilde{X} = 0$. We first calculate the ray of the projection of the four vertex points of the image from Camera 1 onto the plane i. For an image vertex point $\tilde{u}_{1c} \in \mathbb{P}^2$, possible locations of the corresponding 3D point can be calculated by using the back projection equation introduced in Eqn. (4.2) in Section 4.1 as:

$$\tilde{X}_{1c}^{(W)}(\varsigma) = P_1^+ \tilde{u}_{1c} + \varsigma \tilde{c}_1^{(W)} \tag{5.25}$$

where P_1 denotes the camera matrix, $(\cdot)^+$ the pseudoinverse of the matrix, $\tilde{u}_{1c}$ the location of the vertex point in the (virtual) image plane, and $\tilde{c}_1^{(W)} = [(\vec{c}_1^{(W)})^T, 1]^T \in \mathbb{P}^3$ is the homogeneous coordinates of the camera position in the global frame (W). ς is a parameter reflecting the depth of the 3D points on the ray. By solving $\tilde{\gamma}_i^T \tilde{X}_{1c}^{(W)}(\varsigma) = 0$ for ς, we determine the intersection points of the rays with the i-th visible plane. This describes a polygon with four vertices for each plane. Determining the visible part of the surfaces inscribed in these polygons, creates a facetted surface delimited by another polygon. This latter polygon describes the limits of the field of view. The common field of view is obtained by back-projecting the vertices of the latter polygon onto the image plane of Camera 2.

If there are multiple planes, i.e. $M' > 1$, any line l_{ij}, which denotes the intersection line between plane $\tilde{\gamma}_i$ and $\tilde{\gamma}_j$, is a FOV border for those planes. The actual FOV is obtained by combining the FOVs of all planes.

5.3.3 Verification of Baseline-scale Invariance

An important property of our common FOV detection approach is that it is invariant under camera baseline scaling. This property brings the advantage that the method is insensitive to the error in the ranging measurements between the two rovers.

Without loss of generality, we simplify the problem by choosing a reference frame with Camera 1 at the origin and its orientation aligned with the local frame of that camera. As a result, the projection matrix of Camera 1 is $P_1 = K_1 R^T_{(C_1 \to W)}[I| - \vec{c}_1^{(W)}] = K_1[I, \vec{0}]$, in which K_1 denotes the intrinsic camera matrix of Camera 1, $R_{(C_1 \to W)} \in \mathbf{SO}(3)$ the rotation matrix between camera and the world frame, and $\vec{c}_1^{(W)} \in \mathbb{R}^3$ the location of the camera in world frame. The second camera is translated by $t \in \mathbb{R}^3$, and rotated by $R \in \mathbb{R}^{3\times 3}$. Its projection matrix is $P_2 = K_2 R^T_{(C_2 \to W)}[I| - \vec{c}_2^{(W)}] = K_2[R|t]$. The vector $t = -R^T_{(C_2 \to W)}\vec{c}_2^{(W)} - \vec{c}_1^{(W)}$ is the baseline between the two cameras. If the depth of the 3D feature point is not available, the vision measurements are invariant to a scale change of the whole scenario. With the definition $\Lambda = diag(1, 1, 1, s)$, scaling of the baseline length by a factor s changes the projection matrix P_2 to $P_2' = K_2[R, st] = P_2\Lambda$.

Assuming that a 3D point with homogeneous coordinates $\tilde{X}^{(W)} = [X_x, X_y, X_z, 1]^T = [(\vec{X}^{(W)})^T, 1]^T \in \mathbb{P}^3$ is projected on the image planes of both cameras with coordinates $\tilde{u}_1 = [u_{x1}, u_{y1}, 1]^T \in \mathbb{P}^2$ and $\tilde{u}_2 = [u_{x2}, u_{y2}, 1]^T \in \mathbb{P}^2$ respectively, implies that $\tilde{u}_1 = P_1\tilde{X}^{(W)}, \tilde{u}_2 = P_2\tilde{X}^{(W)}$. Utilizing the cross-product constraints that $\tilde{u}_1 \times (P_1\tilde{X}^{(W)}) = 0$ and $\tilde{u}_2 \times (P_2\tilde{X}^{(W)}) = 0$, the unknown 3D coordinates of the point can be triangulated with the 2D coordinates on the two image planes by solving the normal equation

$$A\tilde{X}^{(W)} = \begin{bmatrix} u_{x1}(p_1^3)^T - (p_1^1)^T \\ u_{y1}(p_1^3)^T - (p_1^2)^T \\ u_{x2}(p_2^3)^T - (p_2^1)^T \\ u_{y2}(p_2^3)^T - (p_2^2)^T \end{bmatrix} \tilde{X}^{(W)} = 0 \tag{5.26}$$

in which $(p_i^j)^T$ denotes the j-th row of the projection matrix P_i.

Replacing the baseline by the scaled baseline, implies replacing the terms in Eqn. (5.26) by rows $P_1' = P_1\Lambda = P_1, P_2' = P_2\Lambda$ which leads to the equation

$$A'\tilde{X}^{(W)} = A\Lambda\tilde{X}^{(W)} = 0. \tag{5.27}$$

As a result, if $\tilde{X}^{(W)} = [X_x, X_y, X_z, 1]^T$ solves Eqn. (5.26), Eqn. (5.27) has the solution $\tilde{X}_s^{(W)} = \Lambda^{-1}\tilde{X}^{(W)} = [X_x, X_y, X_z, 1/s]^T$. Therefore, the impact of scaling the baseline length by s is that the triangulated coordinates of the feature points are scaled by $1/s$. The corresponding Cartesian coordinates have the relation

$$\vec{X}_s^{(W)} = [sX_x, sX_y, sX_z]^T = s\vec{X}^{(W)}. \tag{5.28}$$

If we compute a plane equation with three non-collinear points from that plane, the Cartesian coordinates of which are triangulated as $\vec{X}_{s1}^{(W)}, \vec{X}_{s1}^{(W)}$ and $\vec{X}_{s3}^{(W)} \in \mathbb{R}^3$ with the same scaled baseline, the plane can be determined by

$$\tilde{\gamma}_s = \begin{bmatrix} (\vec{X}_{s1}^{(W)} - \vec{X}_{s2}^{(W)}) \times (\vec{X}_{s2}^{(W)} - \vec{X}_{s3}^{(W)}) \\ -(\vec{X}_{s1}^{(W)} \times \vec{X}_{s2}^{(W)})^T \vec{X}_{s3}^{(W)} \end{bmatrix} \in \mathbb{P}^3. \tag{5.29}$$

We can normalize the plane as $\tilde{\gamma}_s = [g^T, 1]^T$. According to Eqn. (5.28), the impact of the scale is $\vec{X}_{si}^{(W)} = s\vec{X}_i^{(W)}$, for $i = 1, 2, 3$. As a result, the plane equation follows that $\tilde{\gamma}_s = \Lambda\tilde{\gamma}$, where $\tilde{\gamma}$ is the plane equation with true scale calculated with points coordinates $\vec{X}_1^{(W)}, \vec{X}_2^{(W)}$ and $\vec{X}_3^{(W)}$.

For point $\tilde{u}_1 = [u_{x1}, u_{y1}, 1]^T$ on the image plane of Camera 1, the back projection results in a ray

$$\tilde{X}^{(W)}(\varsigma) = P_1^{+}\tilde{u}_1 + \varsigma[0, 0, 0, 1]^T \tag{5.30}$$

in which ς is a scalar factor for parameterizing line equation. According to Eqn. (5.32), the ray intersects plane $\tilde{\gamma}$ at

$$\tilde{X}_b^{(W)} = \begin{bmatrix} K_1^{+}\tilde{u}_1 \\ -g^T K_1^{+}\tilde{u}_1 \end{bmatrix} \in \mathbb{P}^3 \tag{5.31}$$

$$\tilde{\gamma}^T \tilde{X}_b^{(W)} = [g^T, 1][(K_1^{+}\tilde{u}_1)^T, \varsigma]^T = 0. \tag{5.32}$$

Similarly the ray intersects the plane $\tilde{\gamma}_s$ at $\tilde{X}_{bs}^{(W)} = [(K_1^{+}\tilde{u}_1)^T, -g^T K_1^{+}\tilde{u}_1/s]^T = \Lambda^{-1}\tilde{X}_b^{(W)}$. Here $\tilde{X}_b^{(W)}$ is the original point coordinates while $\tilde{X}_{bs}^{(W)}$ is the corresponding point on the plane calculated with scaled baseline. The projection of $\tilde{X}_b^{(W)}$ on the image plane of Camera 2 yields $\tilde{u}_2 = P_2\tilde{X}_b^{(W)}$, while $\tilde{X}_{bs}^{(W)}$ is projected to

$$\tilde{u}_{2s} = P_2'\tilde{X}_{bs}^{(W)} = P_2\Lambda\Lambda^{-1}\tilde{X}_b^{(W)} = \tilde{u}_2. \tag{5.33}$$

Therefore, the planes induce the same homography between the two cameras for different baseline scales.

The result indicates that our plane-based common field-of-view detection algorithm is invariant under baseline scaling if and only if we back-project the planes using the same baseline parameters that are utilized to triangulate the feature points. Under this assumption, the method is thus robust against baseline length measurement error.

5.3.4 Simulations of Common Field-of-View Detection

We tested the common FOV detection algorithm from Section 5.3.2 with several rather different pairs of stereo images. The performance was rather similar in all cases. Thus we choose three typical scenario to illustrate the results. The first scenario is a wall with items on the shelf. The second scenario is the corner of a floor and walls. The third scenario uses a pair of stereo images from the NASA Mars rover Sojourner. For the feature extraction, SURF detector and descriptor [57] are used for all three scenarios. RANSAC (RANdom SAmple Cpponsensus) [65] is implemented to reduce the number of outliers, so that the comparison is fair to the approaches in [113] and [114]. The results are illustrated in Fig. 5.18, 5.19, and 5.20.

The highlighted part in the images are the detected common FOV. Yellow dots show the matched feature points. The simulation results indicate that our overlapping detection algorithm performs very well in a wide variety of scenarios.

Table 5.4 compares the values of the overlapping similarity metrics defined by feature point numbers and overlapped pixel numbers. In the "Wall" scenario, 30 feature points are associated with an overlap of 45 percent. On Mars, 24 feature points are associated with an overlap of 70 percent. This shows that the number of matched feature points is not a good measure for the common FOV.

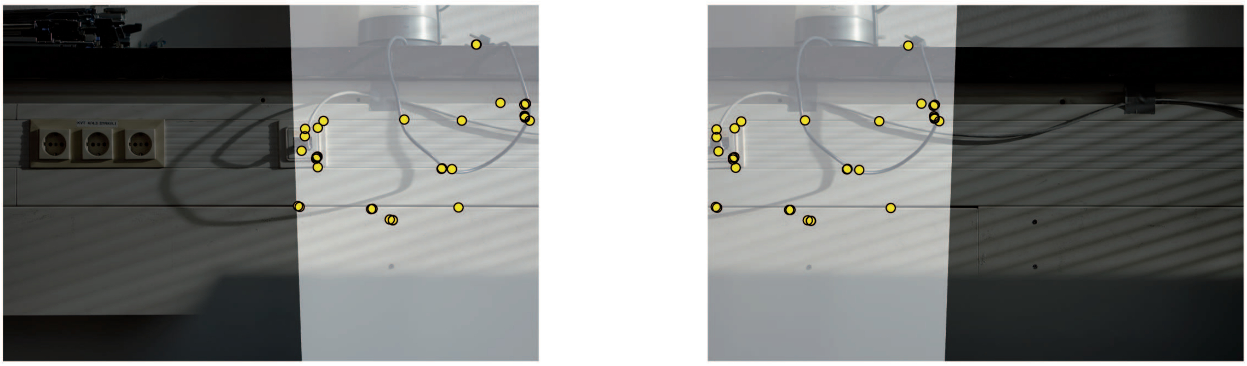


Figure 5.18: Common FOV Detection - Wall

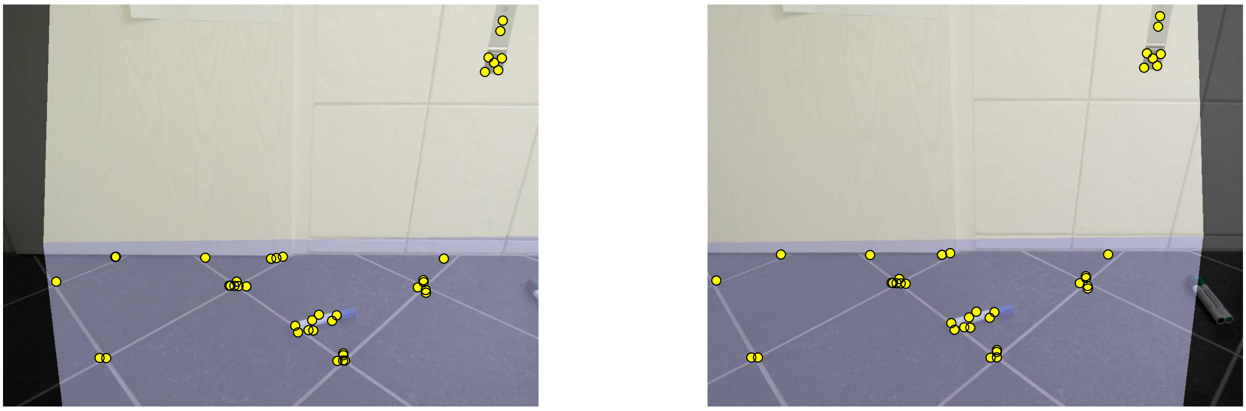


Figure 5.19: Common FOV Detection - Floor

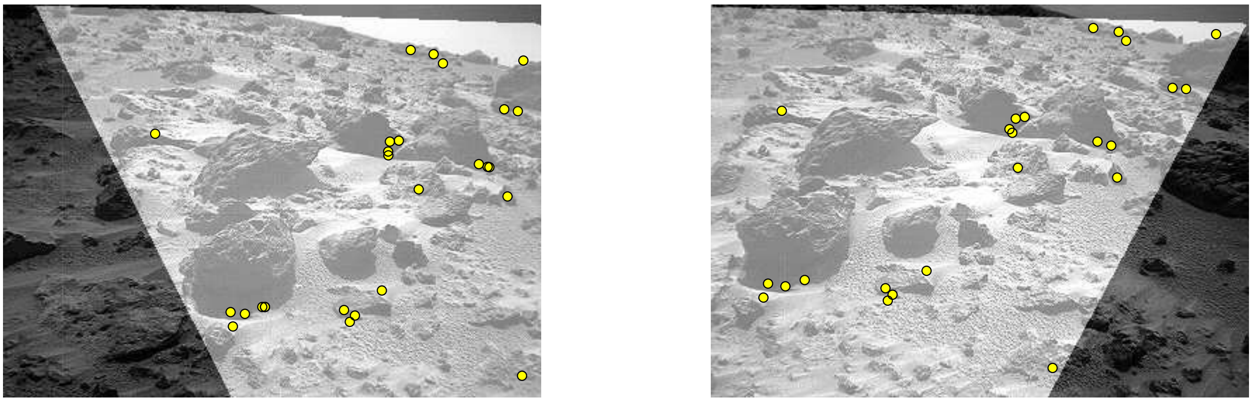


Figure 5.20: Common FOV Detection - Mars

Table 5.4: Common Field-of-View Detection Result

	Detected Plane #	Inlier Points	Overlapped Pixel
Wall	1	30	45.25%
Floor	2	37	91.28%
Mars	1	24	73.50%

6. Visual Navigation of a Cooperative Robotic Swarm

In this chapter, we extend the sensor fusion based relative pose estimation method from Chapter 5 to a multi-robot swarm navigation solution. This approach provides an accurate formation estimate for cooperative SLAM of a robotic swarm by exploiting visual and ranging measurements. Compared with all the state of the art multi-robot SLAM work, our method is based on the relative pose estimation exploiting sensor fusion instead of scene detection and map merging. As a result, neither feature descriptors nor images are required to be transmitted among rovers, so that the communication load is much more feasible in practice. To cope with the scalability problem of swarms with large number of rovers, a grouping algorithm based on inter-rover distance and common field of view is proposed.

6.1 Multi-Agent Visual Navigation–a Review

Inspired by some natural behaviors of animals, the utilization of multi-agent robotic swarms has raised great interests for researchers in recent years [120]. The applications and research of robotic swarms are reviewed in [121]. In exploration tasks, a robotic swarm consisting of several robotic platforms has a number of advantages compared with a single rover. First of all, it is intuitive that the exploration efficiency and flexibility can be significantly improved by using a robotic swarm due to better coverage from multiple independently and simultaneously moving robots. Moreover, a swarm has better observation and perception capability than a single robot. For example, two rovers equipped with monocular camera can form a variant baseline stereo system if the relative pose of them is known. In such cases, the baseline length can be adjusted according to the distance to the observed objects so that the best accuracy can be achieved, and the cameras have mobility to deal with the occlusion problem in 3D reconstruction. In addition, there are usually complicated terrains in exploration, in which a rover may lose its connection to the base station or the orbiter. The connectivity between the rover and the base station can be improved by using a swarm, since other rovers can serve as a relay point, and maintains connection to the anchor points. Last but not least, multiple robots is more robust to individual failures than a single platform. If one of the rovers in the swarm malfunctions, the whole exploration task can still be continued. Due to such advantages, it is interesting to have a globally consistent estimate of the map and the robot poses, so that the robots can avoid collisions, can be navigated autonomously, and extract maximum of information from the measurements.

At the same time, the increase of the robot number also introduces new challenges and problems. One of the most challenging problem is that the egomotion estimation is executed in the local reference frame, i.e., the estimated rover poses and map point locations are relative to the origin of a chosen navigation frame (N). In order to estimate the relative poses and the formation of the swarm elements and to obtain a globally consistent map, information needs to be shared among the robots. Relative pose estimation and the common reference frame determination are two core problems in the swarm navigation. Several approaches have been proposed to solve the multi-robot SLAM problem. In [122], Saeedi et al. reviewed some important SLAM algorithms designed for multi-robot applications. Forster et al. proposed a monocular camera based SLAM approach for multiple UAVs in [27]. The relative pose estimation in the methods is based on map merging, which requires all the UAVs to transmit keyframe poses and local maps to a centralized server. In [26], Fox et al. proposed an approach based on particle filters to map the environment with multiple distributed robot platforms. The egomotion estimation is based on a laser scanner and a wheel odometer on each rover, and the measurements are exchanged between a pair of rovers if they are in communication range. The relative pose between the two rovers is also estimated by merging the local maps. Other map merging based methods include [28, 123, 124].

In [125], Saeedi et al. discussed the multi-robot SLAM problem as a general framework based on particle filters, and divide the situations into the cases before and after the relative pose estimation is successfully executed. As reviewed earlier, the relative pose can be estimated using local map merging using different sensors. However, due to lack of a prior knowledge of a common reference frame in a distributed robotic swarm, the merging based on data association using loop closure detection can become very wrong if mismatch occurs. Indelman et al. remarked the common reference frame determination problem in data association and map merging for multi-robot SLAM, and discussed the inference for the distributed system in [126] and [127].

Another crucial issue in swarm navigation which limits the feasibility of the map merging based methods is the requirements of the communication load. The limited communication has been considered in the approach [26] from Fox et al. and the under water platform based work [128] from Paull et al. In [26], the data transmission is only constrained on close rover pairs. In [128], the inference between the multiple AUVs (Autonomous Underwater Vehicles) are calculated and the robot only transmit the necessary part of the pose graph to each other, which effectively decreases the amount of data needed to be exchanged. However, the relative pose estimation is assumed to be solved in this literature, for which the raw measurements still need to be transmitted using the onboard sensor set-up. Carlone et al. proposed a particle filter based method in [129], which estimates the relative pose by detecting another rover using lidar and vision. This requires rendezvous of two robots with visible line of sight distance. Staudinger et al. proposed a radio based swarm navigation system in GNSS-denied environment in [5]. By exploiting the ranging measurements obtained from the intra-swarm communication links, the relative poses and the formation of the robotic swarm can be estimated in a distributed way with respect to a common reference frame. The agreement of the reference frame converges in a few steps of motion of the swarm. However, the method is based on pure radio but without any other sensor for egomotion estimation and mapping, which has omitted many feasible sensor measurements to improve the performance.

As a viable solution to the aforementioned problems in state-of-the-art approaches, we propose a swarm localization method based on the fusion of visual and ranging measurements. The method is valid as long as the neighboring robots are within radio line-of-sight distance, which is much larger than visually detectable distance. Meanwhile, the communication load is significantly lower than map merging based methods. Generally, the swarm in our set-up can be modeled by a Bayesian network like an example shown in Fig. 6.1. In this instance, the three rovers have communication links among each other. At an arbitrary time instant k, the states of the rovers and the corresponding visual and ranging measurement nodes form a pose graph. Applying the individual visual SLAM algorithm on each rover, the egomotion of the rover can be estimated over time. As a result, the trajectories of all the swarm elements is represented by a large pose graph. The methods for pose graph optimization such as [130] can be applied. It should be mentioned that the egomotion estimation and the mapping using the onboard monocular cameras are referred to the local frames of the rovers and have scale ambiguities. Our approach to merge the nodes from different rovers into a global consistent pose graph will be introduced in the following sections.

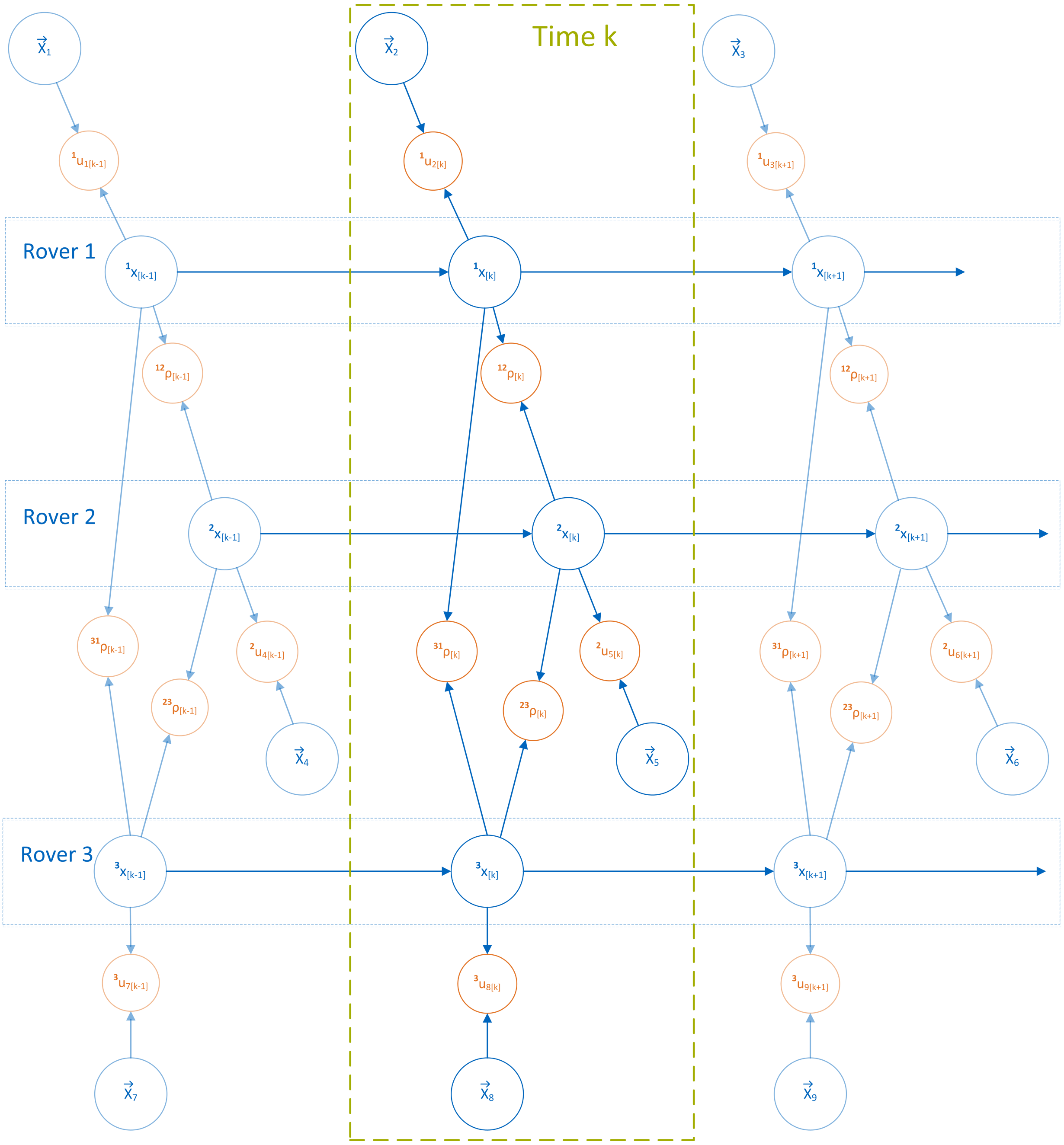


Figure 6.1: Model of the swarm as Bayesian network and pose graph

6.2 Swarm Navigation using Cooperative Vision

The approaches based on map merging and cooperative loop closure detection require to exchange maps from different rovers. However, this demands significant amount of data to be transmitted over the communication channel, since the feature descriptors or the raw visual measurements are essential to merge the map. In many applications, e.g., the Mars exploration, the communication bandwidth and throughput cannot fulfill the requirements. In this section, by applying the relative pose estimation from the ranging and camera fusion introduced in Chapter 5, we propose to estimate the formation of the whole cooperative swarm without transmitting any feature descriptor or raw measurements. In addition, compared with the perception based relative pose estimation methods such as in [129] and [131], our method proposed in Chapter 5 do not

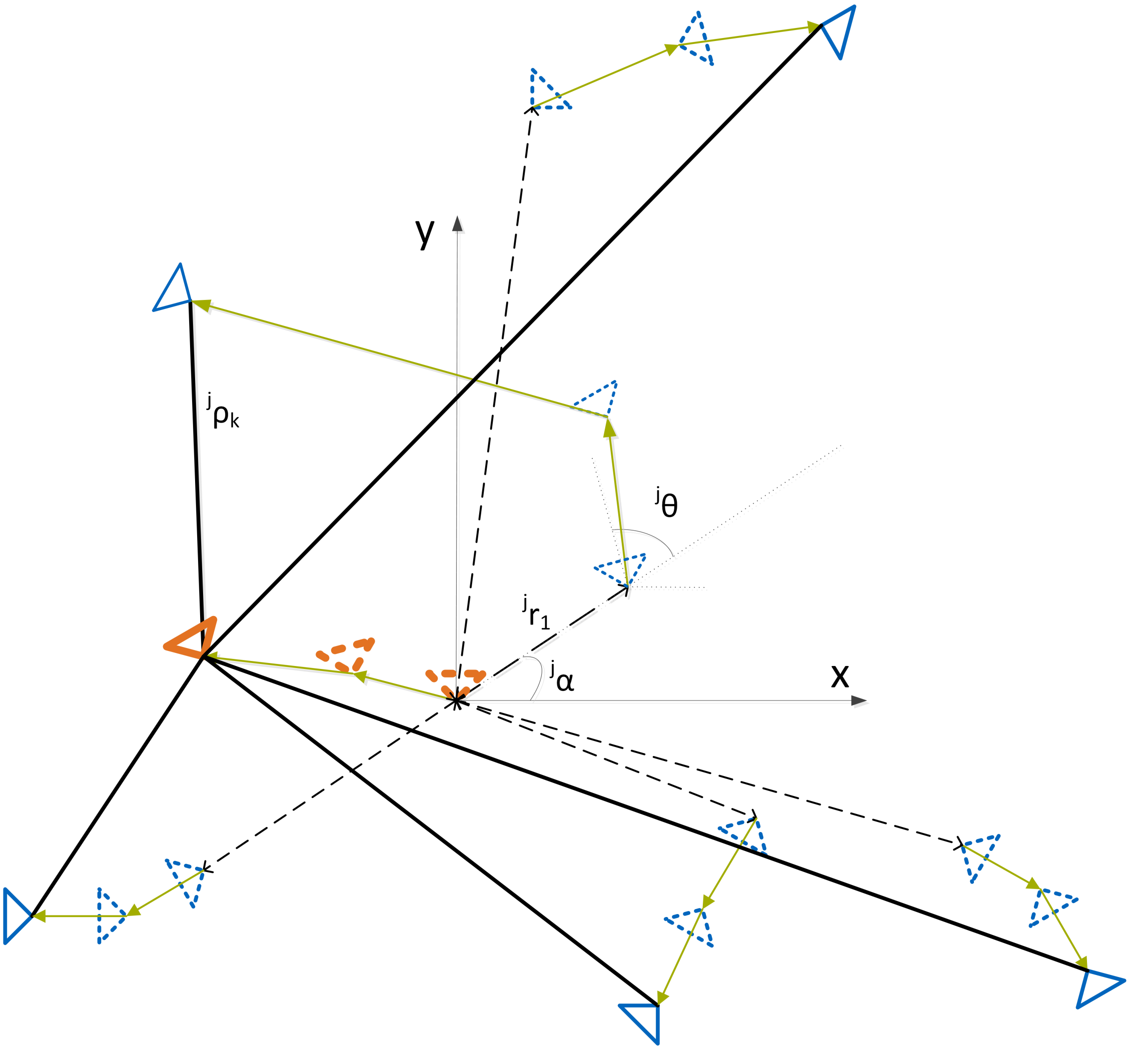


Figure 6.2: Model of the swarm in centralized processing case

require another rover to appear in the field of view of the camera, and the radio line of sight is usually much farther than the effective detection distance of the optical sensors.

We start with a simple centralized approach. Assume among the N_r rovers in the swarm, a central unit with numbering j_1 has connectivity with all the other rovers. The other rovers can transmit their estimated trajectory in the local navigation frame $\{{}^{\mathsf{j}}\hat{\beta}^{(N_j)}_{[k]}|k = 1, ..., N_k, j = j_2, ..., j_{N_r}\}$ to rover j_1. At the same time, the central unit can obtain ranging measurements $\{{}^{\mathsf{j}}\rho_{[k]}|k = 1, ...N_k, j = j_2, ..., j_{N_r}\}$ from the communication links. Fig. 6.2 illustrates the scenario. With the local position estimates and the ranging measurements, the global scale of the rovers ${}^{\mathsf{j_1}}s_g$ as well as ${}^{\mathsf{j}}s_g$ and the relative pose parameters ${}^{\mathsf{j}}\xi = [{}^{\mathsf{j}}r_{[1]}, {}^{\mathsf{j}}\alpha, {}^{\mathsf{j}}\theta]$ between the rover j_1 and all the other rovers j can be estimated by exploiting the scale and relative pose estimation method in Chapter 5. By combining the relative pose between all the trajectories, the formation of the whole swarm can be obtained. This approach is simply an accumulation of estimation result from the pairwise relative pose estimation, which may result in inconsistency of the global scale estimation of the central unit ${}^{\mathsf{j_1}}s_g$.

As an improvement for the swarm navigation, we can generalize the method in Chapter 5 to multi-agent case. We choose the initial pose of the central unit as the reference frame (W). The noisy range measurement between rover j_1 and rover j at time k can be denoted as

$$ {}^{\mathbf{j}}\rho_{[k]} = \left\| {}^{\mathbf{j}}\vec{\beta}_{[k]}^{(W)} - {}^{\mathbf{j_1}}\vec{\beta}_{[k]}^{(W)} \right\| + {}^{\mathbf{j}}\eta_{[k]}. \tag{6.1} $$

The coordinates of the rovers in the reference frame (W) is a function of the global scales and the relative pose parameters that

$$ {}^{\mathbf{j_1}}\vec{\beta}_{[k]}^{(W)} = {}^{\mathbf{j_1}}s_g \, {}^{\mathbf{j_1}}\vec{\beta}_{[k]}^{(N_{j_1})}, \tag{6.2} $$

$$ {}^{\mathbf{j}}\vec{\beta}_{[k]}^{(W)} = {}^{\mathbf{j}}\vec{\beta}_{[1]}^{(W)} + {}^{\mathbf{j}}s_g \, {}^{\mathbf{j}}\vec{\beta}_{[1]}^{(N_j)} = {}^{\mathbf{j}}r_{[1]} R({}^{\mathbf{j}}\alpha)[1,0]^T + {}^{\mathbf{j}}s_g R({}^{\mathbf{j}}\alpha + {}^{\mathbf{j}}\theta - \frac{\pi}{2}) {}^{\mathbf{j}}\vec{\beta}_{[1]}^{(N_j)}. \tag{6.3} $$

Stacking the unknown parameters to a vector $\xi = [{}^{\mathbf{j_1}}s_g, {}^{\mathbf{j_2}}\xi, ..., {}^{\mathbf{j_{N_r}}}\xi]^T$, the true range between rover j_1 and any other rover j can be written as a function of ξ as

$$ {}^{\mathbf{j}}H_k(\xi) = \left\| {}^{\mathbf{j}}\vec{\beta}_{[k]}^{(W)} - {}^{\mathbf{j_1}}\vec{\beta}_{[k]}^{(W)} \right\| = \left\| {}^{\mathbf{j}}r_{[1]} R({}^{\mathbf{j}}\alpha)[1,0]^T + {}^{\mathbf{j}}s_g R({}^{\mathbf{j}}\alpha + {}^{\mathbf{j}}\theta - \frac{\pi}{2}) {}^{\mathbf{j}}\vec{\beta}_{[1]}^{(N_j)} - {}^{\mathbf{j_1}}s_g \, {}^{\mathbf{j_1}}\vec{\beta}_{[k]}^{(N_{j_1})} \right\|. \tag{6.4} $$

Consequently, the scale and relative pose parameters can be estimated by the following optimization:

$$ \hat{\xi} = \arg\min_{\xi} \sum_{j=j_2}^{j_{N_r}} \| {}^{\mathbf{j}}\rho - {}^{\mathbf{j}}H(\xi) \|^2_{\Sigma_{\rho_j}^{-1}}, \quad \text{s.t. } B\xi > 0, \tag{6.5} $$

where ${}^{\mathbf{j}}\rho = [{}^{\mathbf{j}}\rho_{[1]}, {}^{\mathbf{j}}\rho_{[2]}, ..., {}^{\mathbf{j}}\rho_{[N_k]}]^T$, ${}^{\mathbf{j}}H(\xi) = [{}^{\mathbf{j}}H_1(\xi), {}^{\mathbf{j}}H_2(\xi), ..., {}^{\mathbf{j}}H_{N_k}(\xi)]^T$. Here it is assumed that the ranging measurements between the central unit and different other rovers are independent. Σ_{ρ_j} is the covariance matrix of the ranging measurements noise, which is diagonal if the noise is independent of time. Similarly as in Eqn. (5.7), B is a selection matrix to ensure the positiveness of the scale and range parameters. The same as the two rovers case, the problem can be decoupled to an unconstrained optimization and parameters mapping introduced in Section 5.1. The unconstrained optimization can be solved analytically using linearization and iterative optimization algorithm such as Levenberg-Marquart algorithm [111].

$$ \hat{\xi}_{i+1} = \hat{\xi}_i + \left(\sum_{j=j_2}^{j_{N_r}} {}^{\mathbf{j}}J_{\xi}^T(\hat{\xi}_i) \Sigma_{\rho_j}^{-1} {}^{\mathbf{j}}J_{\xi}(\hat{\xi}_i) \right)^{-1} \left(\sum_{j=j_2}^{j_{N_r}} {}^{\mathbf{j}}J_{\xi}^T(\hat{\xi}_i) \Sigma_{\rho_j}^{-1} ({}^{\mathbf{j}}\rho - {}^{\mathbf{j}}H(\hat{\xi}_i)) \right). \tag{6.6} $$

With the above coarse estimation of the parameters vector ξ, the central unit can further fuse the sensor data from the whole cooperative swarm in a tight coupling way to achieve better accuracy, i.e., optimizing the parameters using the raw feature location measurements instead of the estimated trajectories. However, this approach requires all the rovers to transmit their 2D features location and 3D map points location to the central unit.

For a robotic swarm consisting of many rovers, the communication links may not afford the amount of data to be transmitted, even though no feature descriptor is needed to be sent. As a result, we use loose coupling sensor fusion in the swarm navigation. The VSLAM trajectory estimates $\{{}^{\mathbf{j}}\hat{\vec{\beta}}_{[k]}^{(N_j)} | k = 1, ..., N_k, j = j_1, ..., j_{N_r}\}$ obtained with a monocular camera is calculated following a dead reckoning concept. The translation vectors are estimated between keyframes up to a global scale factor, and the poses are propagated using the estimated motion. The rovers can transmit the estimated translation vectors and the corresponding spatial uncertainty information to the other rovers (in the centralized case, to the central unit). For N_k keyframes, the exchanged data from each rover is only N_k two-dimensional vectors and N_k corresponding covariance matrices. The amount of data transmitted is much smaller than the tight coupling approach, which requires to transmit up to $N_m N_k$ 2D vectors for feature locations and N_p 3D vectors for map point locations as well as the corresponding covariance matrices. Consequently, the loose coupling is practically more feasible for the swarm navigation application if real-time onboard processing is demanded.

In order to obtain the trajectory of the whole swarm, it requires to estimate a vector ζ consisting of the unknown parameters as

$$\zeta = \{{}^{\mathbf{j}}\vec{\beta}_{[k]}^{(W)} | k = 1, ..., N_k, j = j_1, ..., j_{N_r}\} \setminus \{{}^{\mathbf{j_1}}\vec{\beta}_{[1]}^{(W)}\} \cup \{{}^{\mathbf{j}}\phi | j = j_2, ..., j_{N_r}\} \in \mathbb{R}^{(2N_rN_k+N_r-3)\times 1}, \quad (6.7)$$

including both the positions of the N_r rovers over N_k keyframes and the relative attitude angles between the navigation frames (N_j) and the common reference frame (W). The angle for rover j is defined as shown in Fig. (6.2) as

$${}^{\mathbf{j}}\phi = {}^{\mathbf{j}}\alpha + {}^{\mathbf{j}}\theta - \frac{\pi}{2}. \quad (6.8)$$

The reference frame is chosen according to the initial pose of the rover j_1, so the position ${}^{\mathbf{j_1}}\vec{\beta}_{[1]}^{(W)} = \vec{0}$ and the angle ${}^{\mathbf{1}}\phi = 0$ are known values and thereby excluded from unknown vector.

The navigation solution using the loosely coupled sensor fusion can be obtained by optimizing

$$\hat{\zeta} = \arg\min_{\zeta} \sum_{j=j_2}^{j_{N_r}} \left\| {}^{\mathbf{j}}\rho - {}^{\mathbf{j}}H(\zeta) \right\|_{\Sigma_{\rho_j}^{-1}}^2 + \sum_{j=j_1}^{j_{N_r}} \sum_{k=2}^{N_k} \left\| {}^{\mathbf{j}}\vec{t}_{[k]}^{(N_j)} - {}^{\mathbf{j}}T_{[k]}(\zeta) \right\|_{\Sigma_{t_{jk}}^{-1}}^2, \quad (6.9)$$

where ${}^{\mathbf{j}}\rho \in \mathbb{R}^{N_k\times 1}, {}^{\mathbf{j}}H(\zeta) \in \mathbb{R}^{N_k\times 1}$ are defined the same as in Eqn. (6.5), except that the entries ${}^{\mathbf{j}}H_k(\zeta)$ is now a function of unknown vector ζ. The latter part of the Eqn. (6.9) contains the information from the translation measurements estimated by the monocular vision. It should be mentioned that the vision based egomotion estimates $\{{}^{\mathbf{j}}\vec{t}_{[k]}^{(N_j)}\}$ are obtained in the navigation frames of the rovers. The translation in the global frame and the navigation frames can be associated by

$${}^{\mathbf{j}}\vec{\beta}_{[k]}^{(W)} - {}^{\mathbf{j}}\vec{\beta}_{[k-1]}^{(W)} = {}^{\mathbf{j}}s_g R({}^{\mathbf{j}}\phi)({}^{\mathbf{j}}\vec{\beta}_{[k]}^{(N_j)} - {}^{\mathbf{j}}\vec{\beta}_{[k-1]}^{(N_j)}) + {}^{\mathbf{j}}\vec{\beta}_{[1]}^{(W)}. \quad (6.10)$$

Consequently, the true translation at time k in the navigation frame of rover j can be written as a function of the unknown parameters in ζ as

$${}^{\mathbf{j}}T_{[k]}(\zeta) = {}^{\mathbf{j}}\vec{\beta}_{[k]}^{(N_j)} - {}^{\mathbf{j}}\vec{\beta}_{[k-1]}^{(N_j)} = \frac{R^T({}^{\mathbf{j}}\phi)({}^{\mathbf{j}}\vec{\beta}_{[k]}^{(W)} - {}^{\mathbf{j}}\vec{\beta}_{[k-1]}^{(W)} - {}^{\mathbf{j}}\vec{\beta}_{[1]}^{(W)})}{\left\| {}^{\mathbf{j}}\vec{\beta}_{[2]}^{(W)} - {}^{\mathbf{j}}\vec{\beta}_{[1]}^{(W)} \right\|}, \quad (6.11)$$

where the global scale is defined as the length of the first translation ${}^{\mathbf{j}}s_g = \|{}^{\mathbf{j}}\vec{\beta}_{[2]}^{(W)} - {}^{\mathbf{j}}\vec{\beta}_{[1]}^{(W)}\|$ as explained in detail in Section 4.1.

In the optimization of Eqn. (6.9), there are $2N_rN_k + N_r - 3$ unknown parameters to be estimated. Meanwhile there exists in total $(3N_r - 1)N_k - 2N_r$ measurements ($2N_r(N_k - 1)$ translation measurements and $(N_r - 1)N_k$ ranging measurements). In order to have sufficient degrees of freedom to solve the equation, it is essential that $(3N_r - 1)N_k - 2N_r \geq 2N_rN_k + N_r - 3$, which can be simplified to the constraint $(N_r - 1)(N_k - 3) \geq 0$. Therefore, at least 3 keyframes for each rover are required. Moreover, the optimization is non-linear, so it demands coarse estimates as initial values to ensure the convergence to the correct optima. Since the coarse global scale and relative pose estimation method in Section 5.1 requires $N_k \geq 5$ keyframes, at least 5 keyframes from each rover are necessary to obtain the swarm formation solution in the centralized mode.

In practice, using centralized processing for a robotic swarm is inefficient. If the number of the rovers N_r is large, the computational power required from the central unit will be too high for real-time processing. Meanwhile, the multiple access efficiency of the communication channel decreases linearly with the number of connected users. Therefore, a decentralized mode is preferred in swarm navigation, i.e., each rover can exchange data and obtain ranging measurements only with its neighbors. If all the rovers have at least one neighbor in its communication range, the swarm can be presented by a connected graph. Fig. 6.3 illustrates

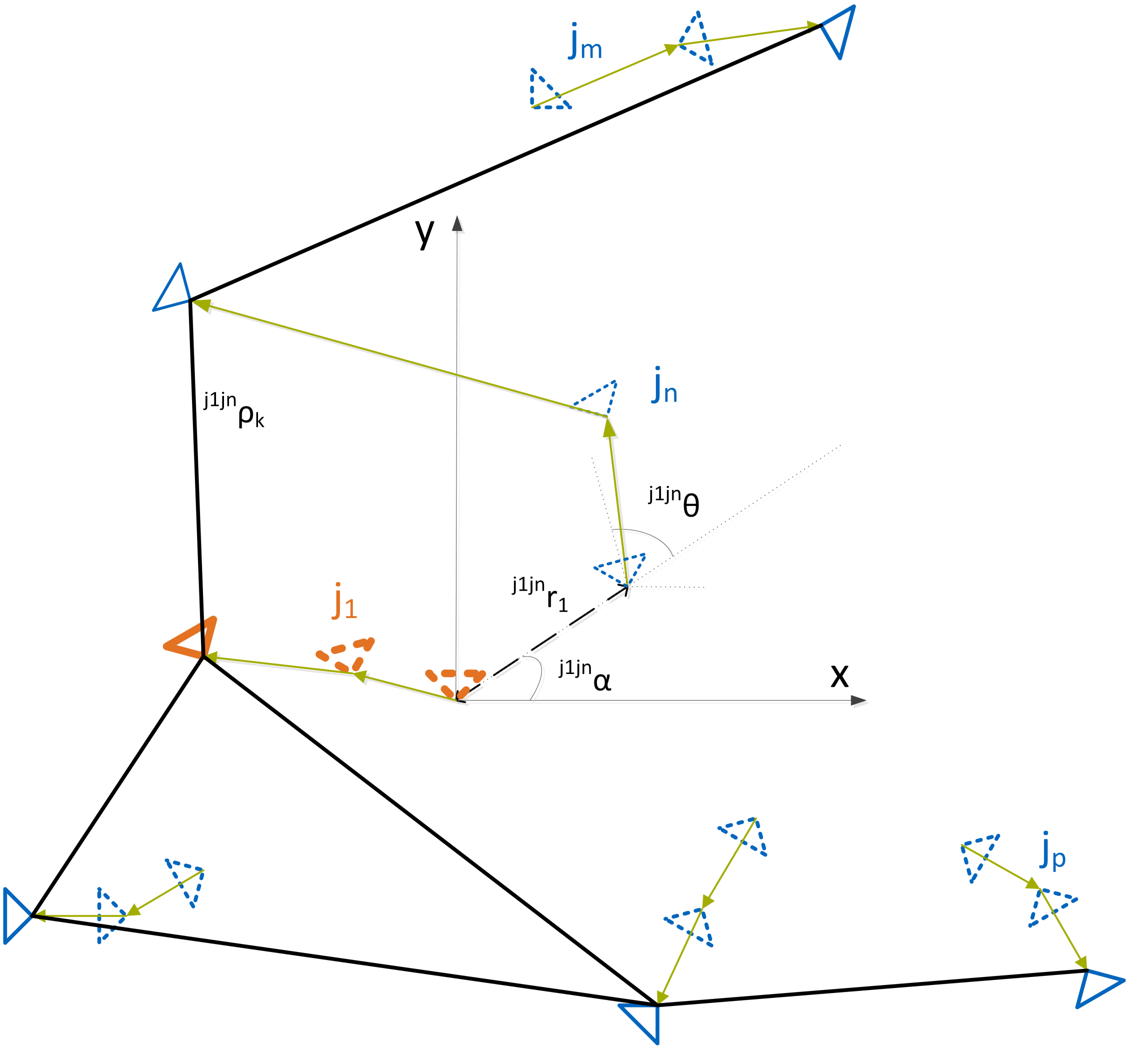


Figure 6.3: Model of the swarm in decentralized processing case

an example of the connectivity for a decentralized swarm. If some rovers are too far away from all the others, one can use grouping strategy to divide the swarm into multiple groups and exploit the algorithm in each group. The grouping strategy will be discussed in Section 6.3. To navigate the swarm, all the rovers should agree on a common global coordinate system in the decentralized mode. Without loss of generality, here we assume that the swarm reference frame (W) is chosen according to the initial pose of the rover j_1, i.e., the orange camera in the example in Fig. 6.3. The coordinates represented in the navigation frame of the reference rover (N_{j_1}) is transformed to that in the swarm reference frame (W) by multiplying a scale factor ${}^{\mathbf{j_1}}s_g$. The transformation of the two reference frames can be expressed by a similarity transformation

$$C_{(N_{j_1}\rightarrow W)} = \begin{bmatrix} s_{(N_{j_1}\rightarrow W)}R_{(N_{j_1}\rightarrow W)} & \vec{t}_{(N_{j_1}\rightarrow W)} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{\mathbf{j_1}}s_g I_3 & \vec{0} \\ 0 & 1 \end{bmatrix} \in \mathbf{Sim}(3). \tag{6.12}$$

Moreover, all the rovers should transmit their local egomotion estimation from monocular VSLAM and the ranging measurements to their neighbor nodes. For an arbitrary rover in the swarm j_n, with the local egomotion estimates and the ranging measurements from its neighbors, the relative pose with any neighbor rover of it can be estimated by exploiting the scale and relative pose estimation method in Chapter 5. Using rover j_n's navigation frame as reference frame, the global scale of the rover j_n and its neighbor j_m, as well as the relative pose between the two rovers in the navigation frame (N_{j_n}), which is parameterized by

${}^{\mathbf{j_n j_m}}\xi = [{}^{\mathbf{j_n j_m}}r_{[1]}, {}^{\mathbf{j_n j_m}}\alpha, {}^{\mathbf{j_n j_m}}\theta]$ here, can be obtained. The coordinates transformation between the two navigation frames can be represented by the following similarity transformation:

$$C_{(N_{jm}\to N_{jn})} = \begin{bmatrix} s_{(N_{jm}\to N_{jn})} R_{(N_{jm}\to N_{jn})} & \vec{t}_{(N_{jm}\to N_{jn})} \\ 0 & 1 \end{bmatrix} \in \mathbf{Sim}(3), \tag{6.13}$$

with

$$s_{(N_{jm}\to N_{jn})} = \frac{{}^{\mathbf{j_m}}s_g}{{}^{\mathbf{j_n}}s_g}, \tag{6.14}$$

$$\vec{t}_{(N_{jm}\to N_{jn})} = \begin{bmatrix} {}^{\mathbf{j_n j_m}}r_{[1]} \cos({}^{\mathbf{j_n j_m}}\alpha) \\ {}^{\mathbf{j_n j_m}}r_{[1]} \sin({}^{\mathbf{j_n j_m}}\alpha) \\ 0 \end{bmatrix}, \tag{6.15}$$

$$R_{(N_{jm}\to N_{jn})} = \begin{bmatrix} \cos({}^{\mathbf{j_n j_m}}\phi) & 0 & \sin({}^{\mathbf{j_n j_m}}\phi) \\ -\sin({}^{\mathbf{j_n j_m}}\phi) & 0 & \cos({}^{\mathbf{j_n j_m}}\phi) \\ 0 & -1 & 0 \end{bmatrix}, \tag{6.16}$$

in which ${}^{\mathbf{j_n j_m}}\phi = {}^{\mathbf{j_n j_m}}\alpha + {}^{\mathbf{j_n j_m}}\theta - \frac{\pi}{2}$.

According to the basic reference frame transformation introduced in Eqn. (2.10), by knowing the scales and the relative poses, the coordinates in the navigation frame (N_{j_m}) can be transformed to that in the world frame by $C_{(N_{j_m}\to W)} = C_{(N_{j_n}\to W)} C_{(N_{j_m}\to N_{j_n})}$. Taking the scenario shown in Fig. 6.4 as an example, the three nodes j_1, j_n, and j_m can be merged into a pose graph in the swarm reference frame (W) by knowing the scale and relative pose estimation results between the two rover pairs.

Therefore, by exchanging the local relative pose estimation and the local knowledge of the swarm formation with all the neighbors, the relative pose information can be merged to obtain a pose graph of the whole formation, if and only if the graph is connected. The propagation delay of the information among swarm elements is proportional to the L1-norm (Manhattan distance) of the longest path between two swarm elements. As the geometric impact, if a node is in the central part of the swarm, the formation pose graph merging will converge fast. On the contrary, for a rover at the edge of the swarm, the distance from the farthest node (measured in L1-norm) will be large, so the knowledge exchange has larger delay. For instance, in Fig. 6.3, the building process of the formation pose graph will be much faster for rover j_1 than that for rover j_m.

Compared with centralized case in Fig. 6.2, there exists loops in the graph since multiple close rovers connect to each other as neighbor nodes. The redundancy provides better accuracy and robustness for the swarm navigation. In an extreme case, if all the rovers are connected to each other, the constructed pose graph will be a complete graph with $N_r(N_r-1)/2$ edges, i.e. the geometry of the swarm is constrained by as much as $N_r(N_r-1)/2$ ranging measurements.

Similarly as in the centralized case, the formation of the swarm can be coarsely estimated by combining the ranging measurements and VSLAM trajectory measurements from all the rovers as:

$$\hat{\zeta} = \arg\min_{\zeta} \sum_{j_m=1}^{N_r} \sum_{j_n=j_m+1}^{N_r} a_{j_n j_m} \| {}^{\mathbf{j_n j_m}}\rho - {}^{\mathbf{j_n j_m}}H(\zeta) \|^2_{\Sigma_\rho^{-1}} + \sum_{j=j_1}^{j_{N_r}} \sum_{k=2}^{N_k} \left\| {}^{\mathbf{j}}\vec{t}^{(N_j)}_{[k]} - {}^{\mathbf{j}}T_{[k]}(\zeta) \right\|^2_{\Sigma^{-1}_{t_{jk}}} \tag{6.17}$$

where $a_{j_n j_m}$ is the connectivity indicator that $a_{j_n j_m} = 1$ if the two rovers are connected in the swarm, otherwise $a_{j_n j_m} = 0$. In Eqn. (6.17), the parameter vector $\zeta = \{{}^{\mathbf{j_1 j}}\phi | j = j_2, ..., j_{N_r}\} \cup \{{}^{\mathbf{j}}\vec{\beta}^{(W)}_{[k]} | k = 1, ..., N_k, j = j_1, ..., j_{N_r}\} \backslash \{{}^{\mathbf{j_1}}\vec{\beta}^{(W)}_{[1]}\} \in \mathbb{R}^{(2N_r N_k + N_r - 3)\times 1}$ consists of $N_r - 1$ attitude angles and $2N_r N_k - 2$ positions. (The reference frame is chosen according to the initial pose of the rover j_1 so the pose ${}^{\mathbf{j_1}}\vec{\beta}^{(W)}_{[1]}$ is

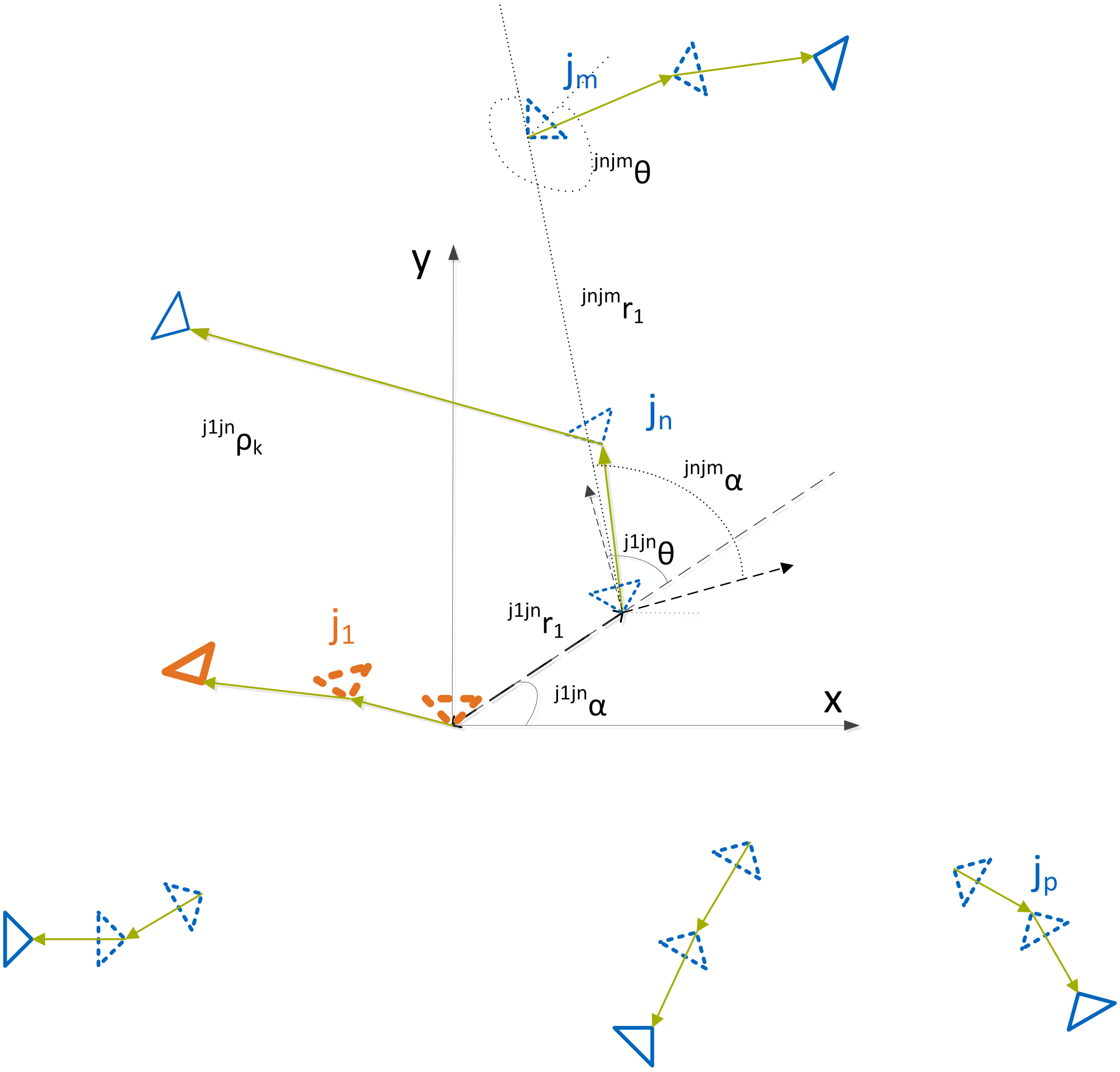


Figure 6.4: Merge of two scale and relative pose estimation results

known.) The vector ${}^{\mathbf{j_n j_m}}\rho \in \mathbb{R}^{N_k \times 1}$ contains the ranging measurements between rover j_m and j_n for all N_k keyframes. The true range function between the two rovers at keyframe N_k is defined as:

$$ {}^{\mathbf{j_n j_m}}H_k(\zeta) = \left\| {}^{\mathbf{j_m}}\vec{\beta}^{(W)}_{[k]} - {}^{\mathbf{j_n}}\vec{\beta}^{(W)}_{[k]} \right\| . \tag{6.18} $$

The translation function for rover j at keyframe N_k is calculated by

$$ {}^{\mathbf{j}}T_{[k]}(\zeta) = {}^{\mathbf{j}}\vec{\beta}^{(N)}_{[k]} - {}^{\mathbf{j}}\vec{\beta}^{(N)}_{[k-1]} = \frac{R^T({}^{\mathbf{j}}\phi)({}^{\mathbf{j}}\vec{\beta}^{(W)}_{[k]} - {}^{\mathbf{j}}\vec{\beta}^{(W)}_{[k-1]} - {}^{\mathbf{j}}\vec{\beta}^{(W)}_{[1]})}{\left\| {}^{\mathbf{j}}\vec{\beta}^{(W)}_{[2]} - {}^{\mathbf{j}}\vec{\beta}^{(W)}_{[1]} \right\|} . \tag{6.19} $$

If there is sufficient communication bandwidth and processing power so that all the rovers can transmit all the visual measurements to the other swarm elements, the measurements can be processed in a tightly coupled way. For large scale pose graph optimization and its convergence, state of the art methods are proposed by Carlone et al. in [132], [133], [134], [135] and [136]. However, in our applications such as autonomous swarm exploration, the requirements are unfeasible, so the loose coupling is preferred as a trade-off between accuracy and processing power for the swarm navigation.

6.3 Autonomous Robotic Grouping exploiting Common Field-of-View [1]

In cooperative swarm exploration, the autonomous navigation of the swarm elements relies on both onboard sensors and cooperative information transferred over intra-swarm communication links. The data transmission requires multiple access and usually has particular requirements on the channel latency and delay to ensure real-time processing of the data. As a result, robots are preferred to be divided into groups to increase the efficiency of exploration and to reduce the communication needs. Moreover, by introducing the groups which consists small number of rovers, the intra-group formation estimation can be realized more efficiently by the centralized or decentralized swarm navigation method introduced in Section 6.2.

The groups are evolving over time due to the movement of the autonomous vehicles. The distance of the robots is normally the dominant consideration in grouping. This is motivated by optimizing intra-group communications and to ease collision avoidance. However, the overlapping FOV of the cameras is also important for swarm VSLAM either for map merging methods in Section 6.1 or for multi-rover stereo reconstruction mentioned in Chapter 5, but it is not considered in most cases. Zou and Tan exploit the number of common feature points in their grouping strategy [137]. However, as we argued in Section 5.3, this is often not a good measure of the overlap of FOVs. Take the common FOV detection results in Fig. 5.18 and 5.20 as an example. In the "Wall" scenario, 30 feature points are correctly associated with a pixel overlapping rate of 45 percent. Meanwhile, for the Mars images, only 24 feature points are matched, but 70 percent of the two images is the common visible part. Therefore, it is more reasonable to apply the percentage of the common FOV than the number of matched feature points as a metric. In this section, we propose a grouping method based on the similarity metric that relies on both common FOV and Euclidean distance. It is obtained from the above adaptive common FOV detection method.

6.3.1 Similarity Metric based on Field-of-View and Distance

Assume that N_r robotic rovers are to be divided into M groups according to a grouping metric, and that the position of the robots ${}^{\mathbf{j}}c^{(W)} \in \mathbb{R}^3$ for $j = 1, 2, ..., N_r$, can be estimated by SLAM and swarm navigation methods, a state-of-the-art choice of a grouping metric d_{ij} is based on distance between rover i and j as in [138]:

$$ d_{ij} = \exp\left(-\frac{\| {}^{\mathbf{i}}\vec{c}^{(W)} - {}^{\mathbf{j}}\vec{c}^{(W)} \|_2^2}{\sigma^2} \right) \tag{6.20} $$

[1] This part of work was joint work with Christoph Bamann, Technische Universität München

with σ being a parameter describing the decay rate of the metric as a function of distance. Close robots are grouped when σ is large. In practice, σ is often chosen as being the standard deviation of the distance among vehicles.

In order to include common FOV in the grouping strategy, we redefine the grouping metric by:

$$s_{ij} = \kappa \cdot (q_i + q_j)/2 + (1 - \kappa) \cdot d_{ij} \tag{6.21}$$

with $q_i = N_{oi}/N_{pi}$, $q_j = N_{oj}/N_{pj}$ being the ratios of the pixel numbers in the overlapping regions and the total pixel number of image i and j, respectively. The parameter $0 \leq \kappa \leq 1$ allows to tune the relative importance of distance and FOV in the grouping strategy. The grouping metric fulfills $0 \leq s_{ij} \leq 1$ and $s_{ii} = 1$.

6.3.2 Autonomous Robots Grouping Using Adaptive Similarity

Exploiting the connectivity between the rovers and the grouping metric in Eqn. (6.21), an undirected graph can be defined to demonstrate the swarm. Let $r_{[1]}, r_{[2]}, ..., r_{[N_r]}$ be the positions of N_r robots, and let them be the vertices of the graph, then connect every vertex-pair $i, j \in \{1, 2, ..., N_r\}$ by an edge with weight s_{ij}. This is a completely connected graph. The M-grouping problem transforms to a graph cut problem that separates the vertices into M disjoint graphs in which the total weight of the remaining edges is maximized. The graph cut problem is NP-hard. Fortunately, excellent suboptimal solutions can be found by using the graph properties and invariants of graph-based matrices. Spectral clustering algorithms such as the Jordan-Weiss algorithm [138] provide good solutions to the M-grouping problem. Spectral clustering uses the property of graph Laplacian matrices that the number of zero eigenvalues in the Laplacian is the number of connected components in the graph. The details of the Laplacian matrix definition and spectral graph theory are introduced in [139]. The Jordan-Weiss algorithm clusters the eigenvectors corresponding to the M largest eigenvalues of the normalized graph Laplacian matrix, which strengthens the cluster property compared with the original data points. In practice, the group number M is usually determined according to the exploration strategy. Nevertheless, if there is a demand to adaptively decide the group number, various criterion can be used to estimate the optimal group number M_{opt}. One option of determining M_{opt} is to maximize the mean intra-group connectivity by

$$M_{opt} = \arg\max_{M} \frac{1}{M} \sum_{i=1}^{M} \lambda_{2,i,M}, \qquad M = 2, ..., M_{max} \tag{6.22}$$

where $\lambda_{2,i,M}$ denotes the algebraic connectivity (or Fiedler value) of the i-th cluster, which is the second smallest eigenvalue of the graph Laplacian of a connected graph. A large Fiedler value implies that the connections among the vertices of the graph are strong, while small Fiedler values indicate that many vertices are connected only by links with a small total weight.

6.3.3 Simulations of Autonomous Grouping Algorithm

The autonomous grouping algorithm is simulated in two scenarios. Scenario 1 contains five cameras with different positions and orientations that distributes as Fig. 6.5.

The robots are grouped using the grouping metric and algorithm introduced in Section 6.3. Assuming that the robots are required to be divided into two groups, different results are obtained when κ is varied, see Table 6.1. For small values of κ , the distance is the dominant, as expected.

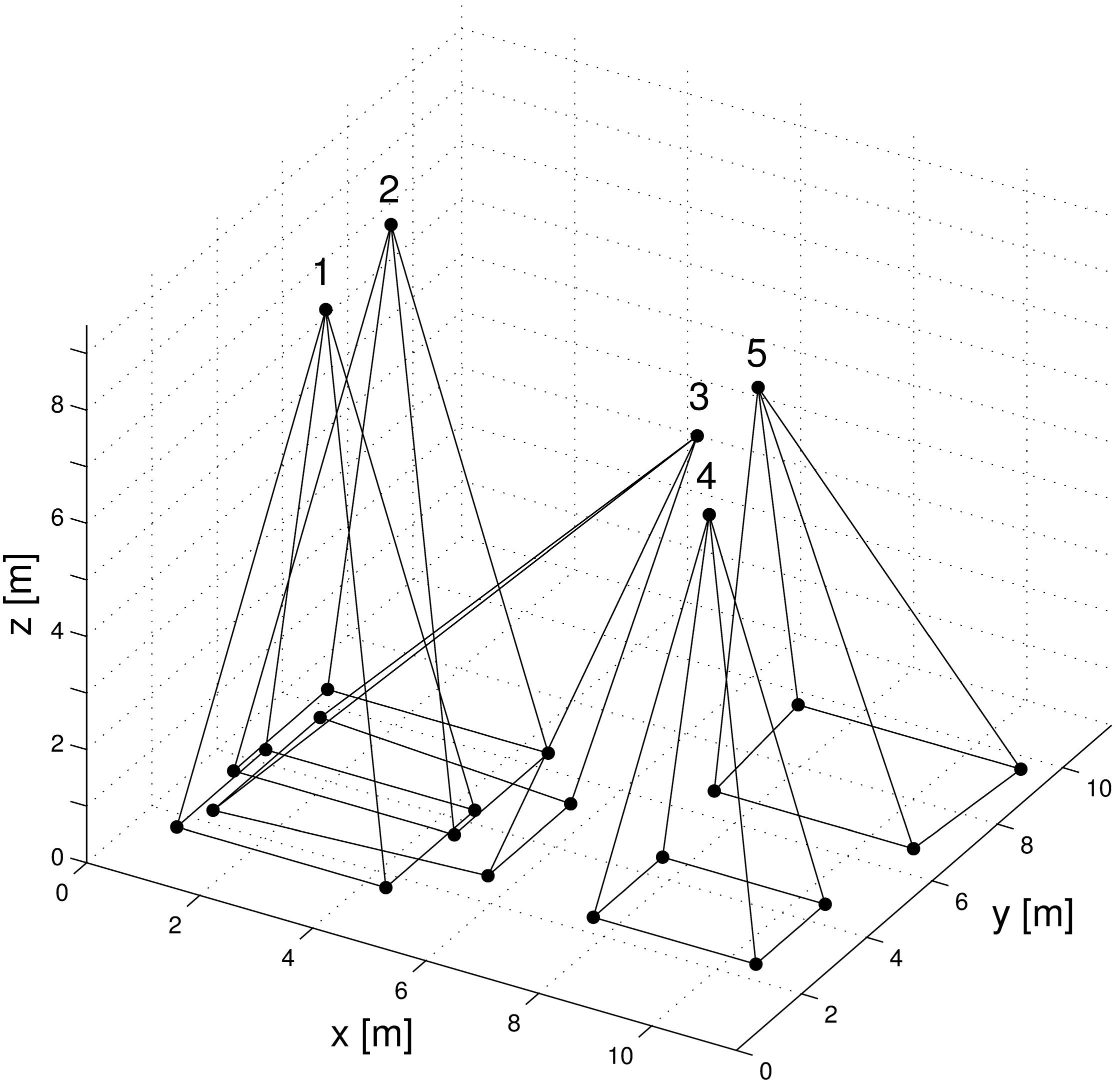


Figure 6.5: Perspective of Scenario 1

Table 6.1: Scenario 1 Grouping Result with $M = 2$

Adaptive Factor	Result Group 1	Result Group 2
$0 \leq \kappa \leq 0.4$	{1,2}	{3,4,5}
$0.5 \leq \kappa \leq 1$	{1,2,3}	{4,5}

Fig. 6.6 illustrates the cameras distribution in Scenario 2. To make the condition more clear, we fix the height of all the robots in the scenario.

Again we cluster the cameras into two distinct groups according to the adaptive similarity metric for various κ values. The grouping result is provided in Table 6.2.

The increase of κ results in a larger impact of common FOV among cameras in the final grouping strategy. To better illustrate it, we plot the average intra-group common FOV and average intra-group distance measure d_{ij} with respect to the incremental adaptive factor κ in Fig. 6.7.

It can be concluded from the curve that for very small κ, the autonomous vehicles with short distance are clustered together. However, the common FOV inside the groups are extremely small. By setting the adaptive factor to a suitable value, the autonomous grouping algorithm results in a good trade-off between

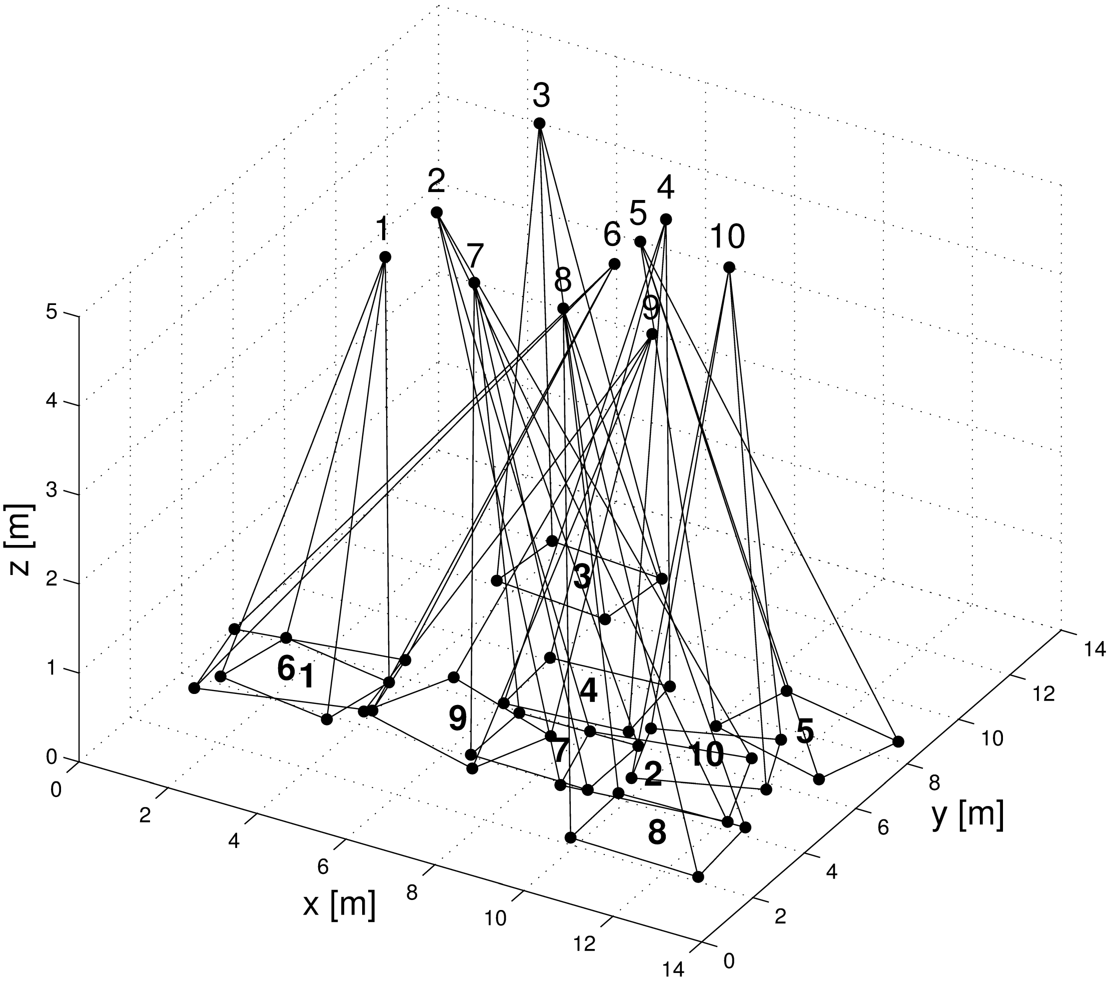


Figure 6.6: Perspective of Scenario 2

Table 6.2: Scenario 2 Grouping Result with $M = 2$

Adaptive Factor	Result Group 1	Result Group 2
$0 \leq \kappa \leq 0.3$	{1,2,3,7}	{4,5,6,8,9,10}
$\kappa = 0.4$	{1,2,3,6,7,8}	{4,5,9,10}
$\kappa = 0.5$	{1,2,6,7,8}	{3,4,5,9,10}
$\kappa = 0.6$	{1,6,7}	{2,3,4,5,8,9,10}
$0.7 \leq \kappa \leq 1$	{1,6}	{2,3,4,5,7,8,9,10}

the robots distance and the common FOV of the cameras. The same conclusion can be drawn for other value of M as well. If M is determined adaptively, the total group number varies for different κ, but the trend remains the same as in Fig. 6.7.

The variation of the autonomous grouping in the situation with vehicle mobility yields the simulation results in Fig. 6.8. In the scenario, the swarm element No. 3 moves along the x-axis while the other elements keep static. The number of the clusters is determined adaptively using the method described in Section 6.3.2. In the simulation, the similarity weighting factor $\kappa = 0.5$, which indicates the distance and the common FOV are treated with equal significance. From Fig. 6.8 we can observe that in the six snapshots the grouping outcome varies as the swarm element moves, and the clusters are marked with different colors. As the swarm element splits from the red group in Fig. 6.8d, 4 clusters are adapted autonomously by the algorithm, and the moving element is clustered into an independent group. When it further approaches

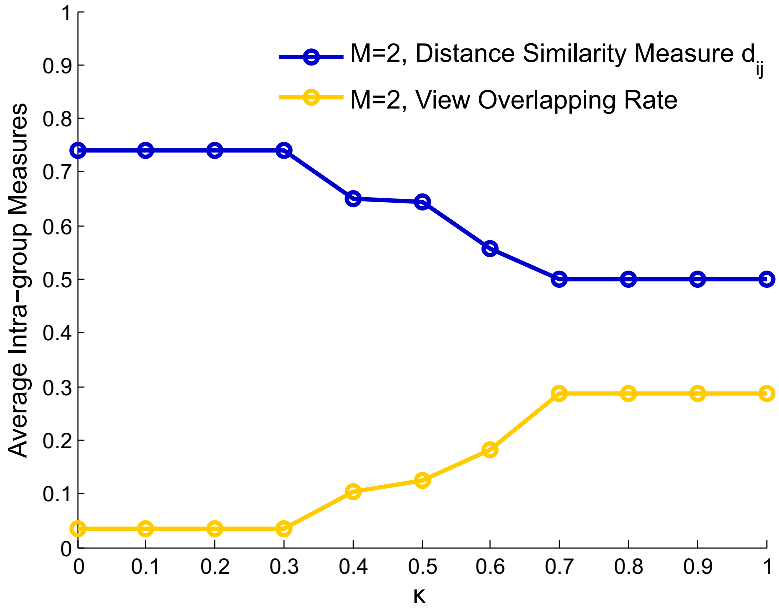
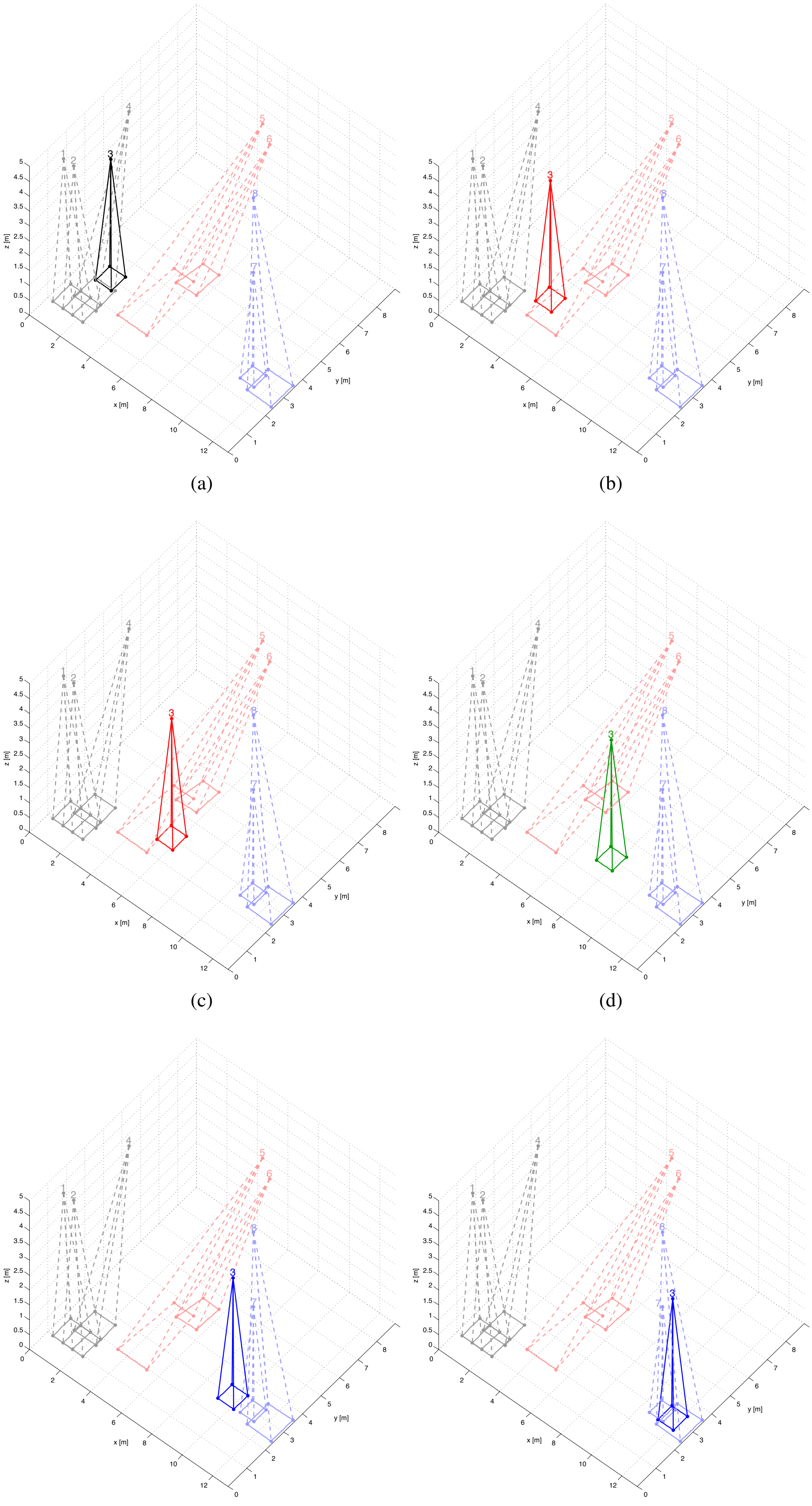


Figure 6.7: Average Intra-group Measures of Scenario 2 over κ

the blue cluster, the moving vehicle is merged into the blue cluster and the group number is again 3. The simulation results indicate reliable grouping and group number adaption performance, and the advantage of using a combination of distance and common FOV as metric can be observed. If only the common FOV is considered, the situation in Fig. 6.8c, Fig. 6.8d and Fig. 6.8e would be the same, which is intuitively not as reasonable as our results.

For the scenario that rover No. 3 moves along x-axis direction, Fig. 6.9 and Fig. 6.10 shows the change of average intra-group common FOV percentage and distance as the rover moves, respectively. The corresponding number of group which is optimized adaptively using Eqn. (6.22) is shown in Fig. 6.11. In the plots, the blue curves are the results from conventional distance-only based grouping, while the orange curves are the results from the proposed adaptive grouping method by considering both distance and common FOV among rovers. In these simulations, the adaptive factor $\kappa = 0.5$. It can be shown that the average intra-group common FOV using the proposed method is always larger than the distance-based method. As a result, if the rovers execute camera-based joint mapping cooperatively with the other rovers inside the group, the adaptive grouping strategy is more preferred. The common FOV advantage is more significant if we increase κ, but the average intra-group distance will also increase, as reflected from Fig. 6.9 and Fig.6.10 for the case $\kappa = 0.7$. The appropriate κ value should be determined according to requirements of the tasks in practice.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.8: Splitting and Merging of a Moving Vehicle

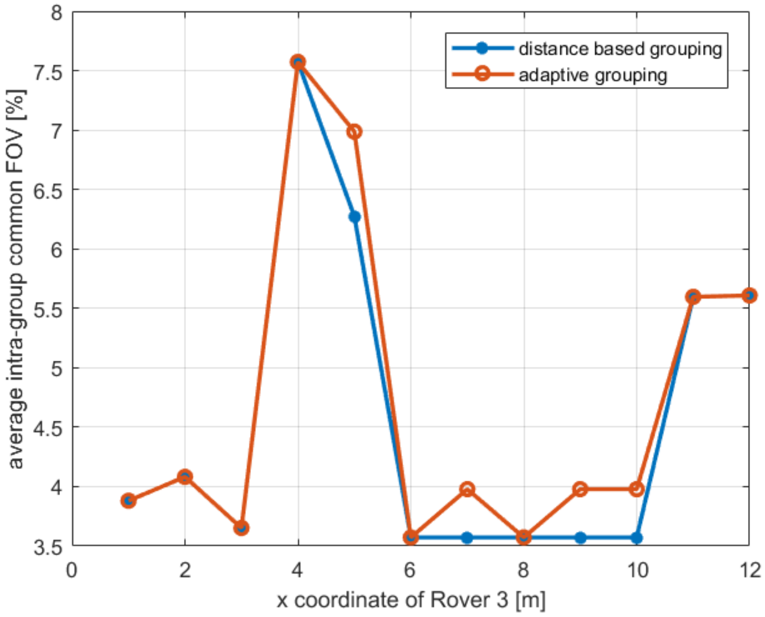


Figure 6.9: Average intra-group common FOV as Rover No. 3 moves, $\kappa = 0.5$

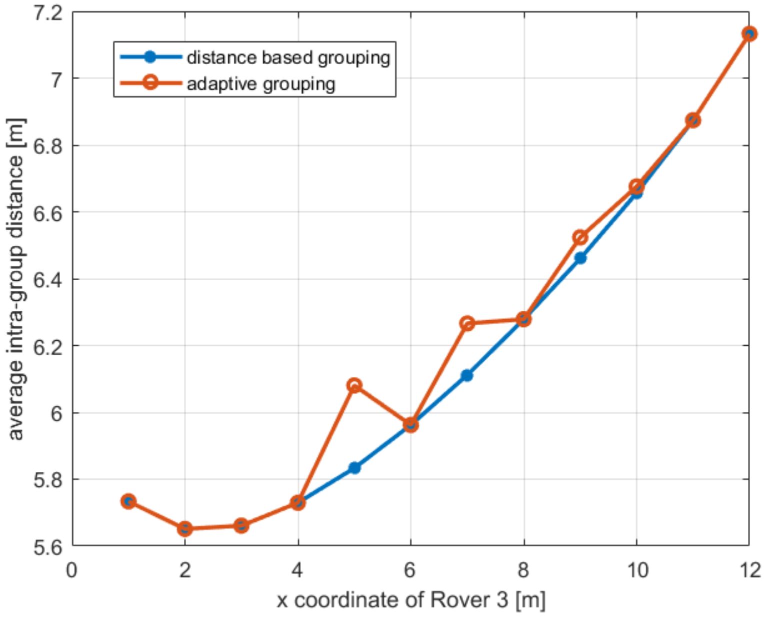


Figure 6.10: Average intra-group distance as Rover No. 3 moves, $\kappa = 0.5$

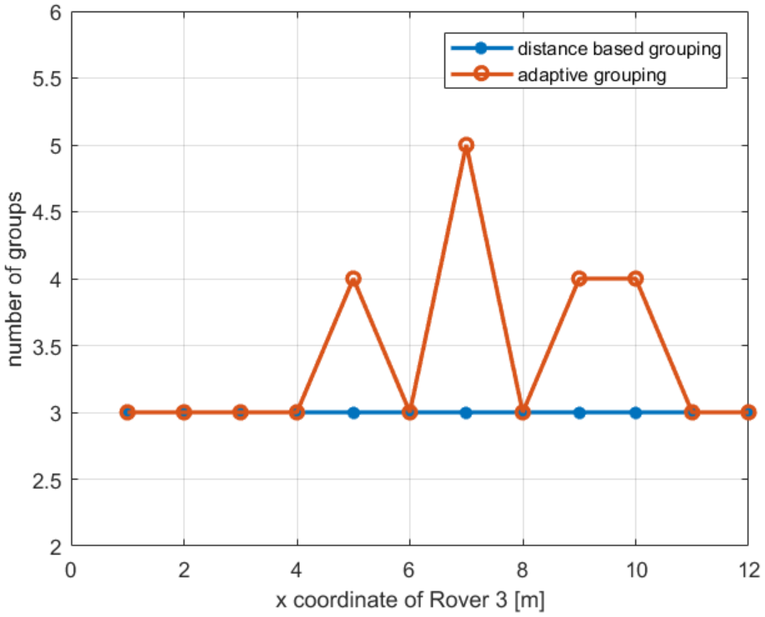


Figure 6.11: Number of groups, $\kappa = 0.5$

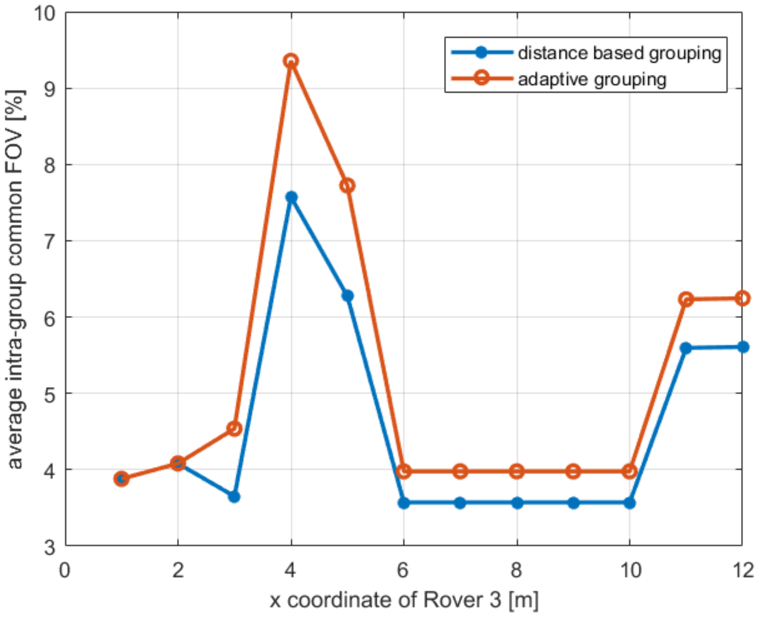


Figure 6.12: Average intra-group common FOV as Rover No. 3 moves, $\kappa = 0.7$

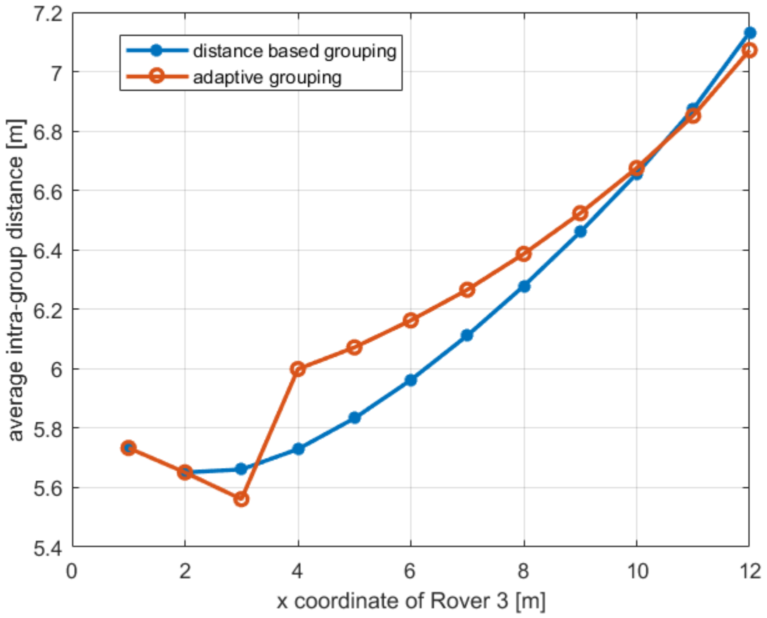


Figure 6.13: Average intra-group distance as Rover No. 3 moves, $\kappa = 0.7$

7. Summary and Conclusions

In conclusion, this dissertation proposed a navigation and localization solution for cooperative robotic swarms in planar motion. By taking the Mars exploration mission as an instance, this work focuses on the situation that no reliable absolute positioning system such as GNSS is available. The sensor measurements are taken from monocular cameras and ranging estimates calculated using pilot signals in communication links. Such sensor set-up is feasible in practice for a robotic swarm, and the cameras as well as basic communications are essential in exploration tasks. Additionally, we have quantitative discussion on the spatial uncertainty of the camera system, which indicates that stereo camera rigs do not have significant advantage over monocular cameras in swarm navigation applications as long as all rovers are connected to at least on other rover for communications and ranging.

We propose sensor fusion methods for rover pairs. One of the rover pair can either be static or dynamic. Both situations are discussed in detail. We have shown that compared with vision-only SLAM methods, the rover pose estimation and the established map accuracy can be improved by tightly coupling the visual and ranging measurements. In addition, the global scale problem in monocular VSLAM can be solved, and the relative pose between the two rovers can be estimated jointly with the scales. As a result, if the pose of one of the rovers is known in the global frame, the other rover can also position itself in the common reference frame. In a degenerated case, if one of the rover is in static mode, the relative pose has a polar angle ambiguity. The methods for resolving the ambiguity has been discussed. The proposed relative pose estimation method has lower requirements on communication capacity compared with state-of-the-art map merging based approaches.

Based on the calculated relative pose, a swarm formation estimation method is proposed. The whole swarm can be modeled as a pose graph. By exchanging local belief of egomotion and relative pose estimation, the pose of the whole swarm can be calculated in a common reference system by pose graph optimization. Both the centralized and distributed strategies are discussed. The solution outperforms conventional multi-robot SLAM approaches in the aspect of data transmission requirements. In the whole calculation, no raw images or feature descriptors are required to be exchanged over communication links.

Moreover, for a pair of rovers with overlapping areas in their cameras' views, 3D reconstruction can be executed as a stereo system with variant baseline. Compared with structure from motion approach using a single camera, the variant baseline stereo system can cope with the reconstruction of scenarios with large scene depth, and has better efficiency in exploration tasks. As a basic step for the swarm-formed stereo vision, an automatic common field of view detection algorithm is proposed. In addition, this dissertation proposed an adaptive swarm grouping method based on both the relative distance and the common field of view between the rovers. By grouping the rovers, the multiple access requirements for the swarm can be relaxed, which makes the formation estimation more feasible.

As an extensive discussion, IMUs are widely used in robotic navigation, and can also provide scale information to monocular camera solutions. On the one hand, a visual-inertial system is yet a dead-reckoning navigation solution. The error accumulation over long time is unbounded. It is still significantly beneficial to introduce the ranging measurements to mitigate the drift. On the other hand, a visual-inertial system can be compatibly integrated to our sensor fusion framework to improve the initialization and estimation accuracy. IMU can be exploited as a complementary sensor to extend the approach in this work.

Appendix

Mathematical notations

The cross product between two three dimensional vectors $\vec{a} = [x_a, y_a, z_a]^T$ and $\vec{b} = [x_b, y_b, z_b]^T$ is calculated as $\vec{a} \times \vec{b} = [y_a z_b - z_a y_b, z_a x_b - x_a z_b, x_a y_b - y_a x_b]^T$. Equivalently, the same result can be generated by matrix multiplication $\vec{a} \times \vec{b} = [a]_\times \vec{b}$ with

$$[a]_\times = \begin{bmatrix} 0 & -z_a & y_a \\ z_a & 0 & -x_a \\ -y_a & x_a & 0 \end{bmatrix}.$$

Note that the mapping from the set of three dimensional skew symmetric matrices to the set of three dimensional vectors is bijective. As a result, any cross product between two 3D vectors can be calculated by constructing a skew symmetric matrix $[\cdot]_\times$ and doing basic matrix multiplication.

Projection function and its Jacobian matrix

Projection from point $\vec{X}_i$ to k-th camera:

$$d_{ik}[\vec{u}_i^{(k)}, 1]^T = K_k[R_{(W\to k)}, t_{(W\to k)}][\vec{X}_i^{(W)}, 1]^T = K_k[R_{(W\to k)}, -R_{(W\to k)}\vec{c}_k^{(W)}][\vec{X}_i^{(W)}, 1]^T$$

d_{ik} is the depth of the point with respect to the camera, i.e.

$$d_{ik} = [0,0,1]K_k R_{(W\to k)}\left(\vec{X}_i^{(W)} - \vec{c}_k^{(W)}\right).$$

$$\vec{u}_{i,x}^{(k)} = \frac{[1,0,0]K_k R_{(W\to k)}\left(\vec{X}_i^{(W)} - \vec{c}_k^{(W)}\right)}{[0,0,1]K_k R_{(W\to k)}\left(\vec{X}_i^{(W)} - \vec{c}_k^{(W)}\right)}$$

$$\vec{u}_{i,y}^{(k)} = \frac{[0,1,0]K_k R_{(W\to k)}\left(\vec{X}_i^{(W)} - \vec{c}_k^{(W)}\right)}{[0,0,1]K_k R_{(W\to k)}\left(\vec{X}_i^{(W)} - \vec{c}_k^{(W)}\right)}$$

In 2D motion case, the pose of the camera can be parameterized by 3 parameters. If the proposed work is extended to 3D scenarios, the camera pose has 6 degrees of freedom (3 from position and 3 from attitude). The 3D attitude of the cameras is described by the special orthonormal group $\mathbf{SO}(3)$, which has different parameterizations, e.g., Euler angle, angle-axis, Lie algebra, and quaternions, etc. As an example, here we give the measurement equations in 3D case using attitude parameterized by unit quaternions. The update of the attitude can either be based on infinitesimal rotations or use a Lagrange multiplier to maintain the unitary constraint in nonlinear optimization. The former solution is widely used in literatures, and the latter solution will be explained in detail in this appendix. The relation between an unit quaternion and its corresponding rotation matrix is

$$R(q) = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2(q_1q_2+q_3q_0) & 2(q_1q_3-q_2q_0) \\ 2(q_1q_2-q_3q_0) & 1-2q_1^2-2q_3^2 & 2(q_2q_3+q_1q_0) \\ 2(q_1q_3+q_2q_0) & 2(q_2q_3-q_1q_0) & 1-2q_1^2-2q_2^2 \end{bmatrix}.$$

If we denote the m-th row and n-th column entry of a matrix by $(\cdot)_{mn}$, we can obtain

$$d_{ik} = R_{31,k}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)}) + R_{32,k}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)}) + R_{33,k}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)}).$$

Here the formula is simplified since the last row of camera intrinsic matrix K_k is always $[0,0,1]$.

$$\vec{u}_{i,x}^{(k)} = \frac{\tilde{x}_{ik}}{d_{ik}}$$

$$\vec{u}_{i,y}^{(k)} = \frac{\tilde{y}_{ik}}{d_{ik}}$$

where

$$\begin{aligned}\tilde{x}_{ik} =& (K_{11}R_{11,k} + K_{12}R_{21,k} + K_{13}R_{31,k})(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)}) \\ &+ (K_{11}R_{12,k} + K_{12}R_{22,k} + K_{13}R_{32,k})(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)}) \\ &+ (K_{11}R_{13,k} + K_{12}R_{23,k} + K_{13}R_{33,k})(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\end{aligned}$$

$$\begin{aligned}\tilde{y}_{ik} =& (K_{21}R_{11,k} + K_{22}R_{21,k} + K_{23}R_{31,k})(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)}) \\ &+ (K_{21}R_{12,k} + K_{22}R_{22,k} + K_{23}R_{32,k})(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)}) \\ &+ (K_{21}R_{13,k} + K_{22}R_{23,k} + K_{23}R_{33,k})(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\end{aligned}$$

The first order partial derivative of the measurements with respect to the parameters are listed below. Using the notation $j = \{1, 2, 3\}$ to represent the matrix entries corresponding to the dimension of $\{x, y, z\}$, it is obvious that

$$\frac{\partial \vec{u}_{i,x}^{(k)}}{\partial \vec{X}_{i,j}^{(W)}} = \frac{\frac{\partial \tilde{x}_{ik}}{\partial \vec{X}_{i,j}^{(W)}} d_{ik} - \tilde{x}_{ik} \frac{\partial d_{ik}}{\partial \vec{X}_{i,j}^{(W)}}}{d_{ik}^2}$$

$$\frac{\partial \vec{u}_{i,y}^{(k)}}{\partial \vec{X}_{i,j}^{(W)}} = \frac{\frac{\partial \tilde{y}_{ik}}{\partial \vec{X}_{i,j}^{(W)}} d_{ik} - \tilde{y}_{ik} \frac{\partial d_{ik}}{\partial \vec{X}_{i,j}^{(W)}}}{d_{ik}^2}$$

with

$$\frac{\partial \tilde{x}_{ik}}{\partial \vec{X}_{i,j}^{(W)}} = K_{11} R_{1j,k} + K_{12} R_{2j,k} + K_{13} R_{3j,k}$$

$$\frac{\partial \tilde{y}_{ik}}{\partial \vec{X}_{i,j}^{(W)}} = K_{21} R_{1j,k} + K_{22} R_{2j,k} + K_{23} R_{3j,k}$$

$$\frac{\partial d_{ik}}{\partial \vec{X}_{i,j}^{(W)}} = R_{3j,k}$$

Similarly, for the camera parameters,

$$\frac{\partial \vec{u}_{i,x}^{(k)}}{\partial \vec{c}_{k,j}^{(W)}} = \frac{\frac{\partial \tilde{x}_{ik}}{\partial \vec{c}_{k,j}^{(W)}} d_{ik} - \tilde{x}_{ik} \frac{\partial d_{ik}}{\partial \vec{c}_{k,j}^{(W)}}}{d_{ik}^2}$$

$$\frac{\partial \vec{u}_{i,y}^{(k)}}{\partial \vec{c}_{k,j}^{(W)}} = \frac{\frac{\partial \tilde{y}_{ik}}{\partial \vec{c}_{k,j}^{(W)}} d_{ik} - \tilde{y}_{ik} \frac{\partial d_{ik}}{\partial \vec{c}_{k,j}^{(W)}}}{d_{ik}^2}$$

with

$$\frac{\partial \tilde{x}_{ik}}{\partial \vec{c}_{k,j}^{(W)}} = -(K_{11} R_{1j,k} + K_{12} R_{2j,k} + K_{13} R_{3j,k})$$

$$\frac{\partial \tilde{y}_{ik}}{\partial \vec{c}_{k,j}^{(W)}} = -(K_{21} R_{1j,k} + K_{22} R_{2j,k} + K_{23} R_{3j,k})$$

$$\frac{\partial d_{ik}}{\partial \vec{c}_{k,j}^{(W)}} = -R_{3j,k}$$

$$\frac{\partial \vec{u}_{i,x}^{(k)}}{\partial q_{k,j}} = \frac{\frac{\partial \tilde{x}_{ik}}{\partial q_{k,j}} d_{ik} - \tilde{x}_{ik} \frac{\partial d_{ik}}{\partial q_{k,j}}}{d_{ik}^2}$$

$$\frac{\partial \vec{u}_{i,y}^{(k)}}{\partial q_{k,j}} = \frac{\frac{\partial \tilde{y}_{ik}}{\partial q_{k,j}} d_{ik} - \tilde{y}_{ik} \frac{\partial d_{ik}}{\partial q_{k,j}}}{d_{ik}^2}$$

with

$$\begin{aligned}\frac{\partial \tilde{x}_{ik}}{\partial q_{k,0}} =& K_{12}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{21,k}}{\partial q_{k,0}} + K_{13}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{31,k}}{\partial q_{k,0}}\\ &+ K_{11}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{12,k}}{\partial q_{k,0}} + K_{13}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{32,k}}{\partial q_{k,0}}\\ &+ K_{11}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{13,k}}{\partial q_{k,0}} + K_{12}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{23,k}}{\partial q_{k,0}}\end{aligned}$$

$$\begin{aligned}\frac{\partial \tilde{x}_{ik}}{\partial q_{k,1}} =& K_{12}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{21,k}}{\partial q_{k,1}} + K_{13}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{31,k}}{\partial q_{k,1}}\\ &+ K_{11}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{12,k}}{\partial q_{k,1}} + K_{12}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{22,k}}{\partial q_{k,1}} + K_{13}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{32,k}}{\partial q_{k,1}}\\ &+ K_{11}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{13,k}}{\partial q_{k,1}} + K_{12}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{23,k}}{\partial q_{k,1}} + K_{13}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{33,k}}{\partial q_{k,1}}\end{aligned}$$

$$\begin{aligned}\frac{\partial \tilde{x}_{ik}}{\partial q_{k,2}} =& K_{11}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{11,k}}{\partial q_{k,2}} + K_{12}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{21,k}}{\partial q_{k,2}} + K_{13}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{31,k}}{\partial q_{k,2}}\\ &+ K_{11}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{12,k}}{\partial q_{k,2}} + K_{13}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{32,k}}{\partial q_{k,2}}\\ &+ K_{11}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{13,k}}{\partial q_{k,2}} + K_{12}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{23,k}}{\partial q_{k,2}} + K_{13}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{33,k}}{\partial q_{k,2}}\end{aligned}$$

$$\begin{aligned}\frac{\partial \tilde{x}_{ik}}{\partial q_{k,3}} =& K_{11}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{11,k}}{\partial q_{k,3}} + K_{12}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{21,k}}{\partial q_{k,3}} + K_{13}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{31,k}}{\partial q_{k,3}}\\ &+ K_{11}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{12,k}}{\partial q_{k,3}} + K_{12}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{22,k}}{\partial q_{k,3}} + K_{13}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{32,k}}{\partial q_{k,3}}\\ &+ K_{11}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{13,k}}{\partial q_{k,3}} + K_{12}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{23,k}}{\partial q_{k,3}}\end{aligned}$$

and

$$\begin{aligned}\frac{\partial \tilde{y}_{ik}}{\partial q_{k,0}} =& K_{22}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{21,k}}{\partial q_{k,0}} + K_{23}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{31,k}}{\partial q_{k,0}}\\ &+ K_{21}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{12,k}}{\partial q_{k,0}} + K_{23}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{32,k}}{\partial q_{k,0}}\\ &+ K_{21}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{13,k}}{\partial q_{k,0}} + K_{22}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{23,k}}{\partial q_{k,0}}\end{aligned}$$

$$\begin{aligned}\frac{\partial \tilde{y}_{ik}}{\partial q_{k,1}} =& K_{22}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{21,k}}{\partial q_{k,1}} + K_{23}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{31,k}}{\partial q_{k,1}}\\ &+ K_{21}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{12,k}}{\partial q_{k,1}} + K_{22}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{22,k}}{\partial q_{k,1}} + K_{23}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{32,k}}{\partial q_{k,1}}\\ &+ K_{21}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{13,k}}{\partial q_{k,1}} + K_{22}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{23,k}}{\partial q_{k,1}} + K_{23}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{33,k}}{\partial q_{k,1}}\end{aligned}$$

$$\begin{aligned}\frac{\partial \tilde{y}_{ik}}{\partial q_{k,2}} =& K_{21}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{11,k}}{\partial q_{k,2}} + K_{22}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{21,k}}{\partial q_{k,2}} + K_{23}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{31,k}}{\partial q_{k,2}} \\ &+ K_{21}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{12,k}}{\partial q_{k,2}} + K_{23}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{32,k}}{\partial q_{k,2}} \\ &+ K_{21}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{13,k}}{\partial q_{k,2}} + K_{22}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{23,k}}{\partial q_{k,2}} + K_{23}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{33,k}}{\partial q_{k,2}}\end{aligned}$$

$$\begin{aligned}\frac{\partial \tilde{y}_{ik}}{\partial q_{k,3}} =& K_{21}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{11,k}}{\partial q_{k,3}} + K_{22}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{21,k}}{\partial q_{k,3}} + K_{23}(\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{31,k}}{\partial q_{k,3}} \\ &+ K_{21}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{12,k}}{\partial q_{k,3}} + K_{22}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{22,k}}{\partial q_{k,3}} + K_{23}(\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{32,k}}{\partial q_{k,3}} \\ &+ K_{21}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{13,k}}{\partial q_{k,3}} + K_{22}(\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{23,k}}{\partial q_{k,3}}\end{aligned}$$

$$\frac{\partial d_{ik}}{\partial q_{k,j}} = (\vec{X}_{i,x}^{(W)} - \vec{c}_{k,x}^{(W)})\frac{\partial R_{31,k}}{\partial q_{k,j}} + (\vec{X}_{i,y}^{(W)} - \vec{c}_{k,y}^{(W)})\frac{\partial R_{32,k}}{\partial q_{k,j}} + (\vec{X}_{i,z}^{(W)} - \vec{c}_{k,z}^{(W)})\frac{\partial R_{33,k}}{\partial q_{k,j}}$$

The Jacobian matrix of the measurements with respect to the parameters would be

$$J_{u\xi} = \begin{bmatrix} \frac{\partial \vec{u}_{i,x}^{(k)}}{\partial q_{k,1}} & \frac{\partial \vec{u}_{i,x}^{(k)}}{\partial q_{k,2}} & \frac{\partial \vec{u}_{i,x}^{(k)}}{\partial q_{k,3}} & \frac{\partial \vec{u}_{i,x}^{(k)}}{\partial \vec{c}_{k,x}^{(W)}} & \frac{\partial \vec{u}_{i,x}^{(k)}}{\partial \vec{c}_{k,y}^{(W)}} & \frac{\partial \vec{u}_{i,x}^{(k)}}{\partial \vec{c}_{k,z}^{(W)}} & \frac{\partial \vec{u}_{i,x}^{(k)}}{\partial \vec{X}_{i,x}^{(W)}} & \frac{\partial \vec{u}_{i,x}^{(k)}}{\partial \vec{X}_{i,y}^{(W)}} & \frac{\partial \vec{u}_{i,x}^{(k)}}{\partial \vec{X}_{i,z}^{(W)}} \\ \frac{\partial \vec{u}_{i,y}^{(k)}}{\partial q_{k,1}} & \frac{\partial \vec{u}_{i,y}^{(k)}}{\partial q_{k,2}} & \frac{\partial \vec{u}_{i,y}^{(k)}}{\partial q_{k,3}} & \frac{\partial \vec{u}_{i,y}^{(k)}}{\partial \vec{c}_{k,x}^{(W)}} & \frac{\partial \vec{u}_{i,y}^{(k)}}{\partial \vec{c}_{k,y}^{(W)}} & \frac{\partial \vec{u}_{i,y}^{(k)}}{\partial \vec{c}_{k,z}^{(W)}} & \frac{\partial \vec{u}_{i,y}^{(k)}}{\partial \vec{X}_{i,x}^{(W)}} & \frac{\partial \vec{u}_{i,y}^{(k)}}{\partial \vec{X}_{i,y}^{(W)}} & \frac{\partial \vec{u}_{i,y}^{(k)}}{\partial \vec{X}_{i,z}^{(W)}} \end{bmatrix}$$

In the iteration, for the attitude parametrization as unit quaternions, only the q_1, q_2, q_3 entries are updated and $q_0 = \pm\sqrt{1 - q_1^2 - q_2^2 - q_3^2}$. $q_0 > 0$ means the rotation is between $-\pi/2$ and $\pi/2$, which is reasonable for the update or refinement if the motion is not so much. Therefore the q_0 can be kept the same sign as the initial attitude before refinement. In the derivative of elements of R matrix, the q_0 must be treated as a function of q_j for j=1,2,3, so

$$\frac{\partial q_0}{\partial q_j} = -\frac{q_j}{q_0}.$$

The derivatives of a rotation matrix with respect to the three quaternion entries q_1, q_2, q_3 are calculated by:

$\frac{\partial R_{11}}{\partial q_1} = 0$	$\frac{\partial R_{11}}{\partial q_2} = -4q_2$	$\frac{\partial R_{11}}{\partial q_3} = -4q_3$
$\frac{\partial R_{12}}{\partial q_1} = 2q_2 + 2q_3\frac{\partial q_0}{\partial q_1}$	$\frac{\partial R_{12}}{\partial q_2} = 2q_1 + 2q_3\frac{\partial q_0}{\partial q_2}$	$\frac{\partial R_{12}}{\partial q_3} = 2q_0 + 2q_3\frac{\partial q_0}{\partial q_3}$
$\frac{\partial R_{13}}{\partial q_1} = 2q_3 - 2q_2\frac{\partial q_0}{\partial q_1}$	$\frac{\partial R_{13}}{\partial q_2} = -2q_0 - 2q_2\frac{\partial q_0}{\partial q_2}$	$\frac{\partial R_{13}}{\partial q_3} = 2q_1 - 2q_2\frac{\partial q_0}{\partial q_3}$
$\frac{\partial R_{21}}{\partial q_1} = 2q_2 - 2q_3\frac{\partial q_0}{\partial q_1}$	$\frac{\partial R_{21}}{\partial q_2} = 2q_1 - 2q_3\frac{\partial q_0}{\partial q_2}$	$\frac{\partial R_{21}}{\partial q_3} = -2q_0 - 2q_3\frac{\partial q_0}{\partial q_3}$
$\frac{\partial R_{22}}{\partial q_1} = -4q_1$	$\frac{\partial R_{22}}{\partial q_2} = 0$	$\frac{\partial R_{22}}{\partial q_3} = -4q_3$
$\frac{\partial R_{23}}{\partial q_1} = 2q_0 + 2q_1\frac{\partial q_0}{\partial q_1}$	$\frac{\partial R_{23}}{\partial q_2} = 2q_3 + 2q_1\frac{\partial q_0}{\partial q_2}$	$\frac{\partial R_{23}}{\partial q_3} = 2q_2 + 2q_1\frac{\partial q_0}{\partial q_3}$
$\frac{\partial R_{31}}{\partial q_1} = 2q_3 + 2q_2\frac{\partial q_0}{\partial q_1}$	$\frac{\partial R_{31}}{\partial q_2} = 2q_0 + 2q_2\frac{\partial q_0}{\partial q_2}$	$\frac{\partial R_{31}}{\partial q_3} = 2q_1 + 2q_2\frac{\partial q_0}{\partial q_3}$
$\frac{\partial R_{32}}{\partial q_1} = -2q_0 - 2q_1\frac{\partial q_0}{\partial q_1}$	$\frac{\partial R_{32}}{\partial q_2} = 2q_3 - 2q_1\frac{\partial q_0}{\partial q_2}$	$\frac{\partial R_{32}}{\partial q_3} = 2q_2 - 2q_1\frac{\partial q_0}{\partial q_3}$
$\frac{\partial R_{33}}{\partial q_1} = -4q_1$	$\frac{\partial R_{33}}{\partial q_2} = -4q_2$	$\frac{\partial R_{33}}{\partial q_3} = 0$

Bibliography

[1] M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the mars exploration rovers,” *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[2] A. S. McEwen, M. C. Malin, M. H. Carr, and W. K. Hartmann, “Voluminous volcanism on early mars revealed in valles marineris,” *Nature*, vol. 397, no. 6720, pp. 584–586, 1999.

[3] S. Sand, S. Zhang, M. Mühlegg, G. Falconi, C. Zhu, T. Krüger, and S. Nowak, “Swarm exploration and navigation on Mars,” in *International Conference on Localization and GNSS, Torino, Italy*, 2013.

[4] “Vamex-cosmic - swarm exploration on mars,” http://www.dlr.de/kn/desktopdefault.aspx/tabid-2081/6941_read-46646/, 2015.

[5] E. Staudinger, S. Zhang, A. Dammann, and C. Zhu, “Towards a radio-based swarm navigation system on mars - key technologies and performance assessment,” in *Wireless for Space and Extreme Environments (WiSEE), 2014 IEEE International Conference on*, Oct 2014, pp. 1–7.

[6] C. Zhu, S. Zhang, A. Dammann, S. Sand, P. Henkel, and C. Günther, “Return-to-base navigation of robotic swarms in mars exploration using doa estimation,” in *Proceedings ELMAR-2013*, Sep. 2013, pp. 349–352.

[7] S. Zhang, R. Raulefs, and A. Dammann, “Location information driven formation control for swarm return-to-base application,” in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug. 2016, pp. 758–763.

[8] T. Wiedemann, C. Manss, and D. Shutin, “Multi-agent exploration of spatial dynamical processes under sparsity constraints,” *Autonomous Agents and Multi-Agent Systems*, vol. 32, no. 1, pp. 134–162, Jan 2018.

[9] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, and L. Meier, “Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in gps-denied environments,” *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 26–40, Sept 2014.

[10] *Study on LTE device to device proximity services; Radio aspects*, 3GPP, 3 2014, release 12.

[11] *IEEE 802.11ai-2016 - IEEE Standard for Information technology*, IEEE, 12 2016.

[12] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, “Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 86–92, May 2014.

[13] C. A. Poynton, “Rehabilitation of gamma,” in *Human Vision and Electronic Imaging III*, vol. 3299. International Society for Optics and Photonics, 1998, pp. 232–250.

[14] A. Ford and A. Roberts, “Colour space conversions,” *Westminster University, London*, vol. 1998, pp. 1–31, 1998.

[15] J. G. Fryer and D. C. Brown, “Lens distortion for close-range photogrammetry,” *Photogrammetric engineering and remote sensing*, vol. 52, no. 1, pp. 51–58, 1986.

[16] J. Heikkila and O. Silven, “A four-step camera calibration procedure with implicit image correction,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun 1997, pp. 1106–1112.

[17] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov 2000.

[18] E. Eade, “Lie groups for 2d and 3d transformations,” *URL http://ethaneade. com/lie. pdf, revised Dec*, 2013.

[19] R. I. Hartley and P. Sturm, “Triangulation,” *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146 – 157, 1997.

[20] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[21] J. Weng, P. Cohen, and M. Herniou, “Camera calibration with distortion models and accuracy evaluation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, Oct 1992.

[22] Y. Furukawa and J. Ponce, “Accurate camera calibration from multi-view stereo and bundle adjustment,” *International Journal of Computer Vision*, vol. 84, no. 3, pp. 257–268, Sep 2009.

[23] Itseez, “Open source computer vision library,” https://github.com/itseez/opencv, 2015.

[24] J.-Y. Bouguet, “Camera calibration toolbox for matlab,” http://www.vision.caltech.edu/bouguetj/calib_doc/, 2015.

[25] C. Loop and Z. Zhang, “Computing rectifying homographies for stereo vision,” in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 1, 1999, p. 131 Vol. 1.

[26] R. Vincent, D. Fox, J. Ko, K. Konolige, B. Limketkai, B. Morisset, C. Ortiz, D. Schulz, and B. Stewart, “Distributed multirobot exploration, mapping, and task allocation,” *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 229–255, 2008.

[27] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, “Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles,” *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, vol. 143607, no. 200021, pp. 3963–3970, 2013.

[28] E. Montijano and C. Sagues, “Distributed multi-camera visual mapping using topological maps of planar regions,” *Pattern Recognition*, vol. 44, no. 7, pp. 1528–1539, 2011.

[29] M. D. Shuster, “A Survey of attitude representations,” vol. 1, no. 4, pp. 439–517, 1993.

[30] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, March 2018.

[31] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “Dtam: Dense tracking and mapping in real-time,” in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2320–2327.

[32] J. Engel, J. Sturm, and D. Cremers, “Semi-dense visual odometry for a monocular camera,” in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 1449–1456.

[33] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *Computer Vision - ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, vol. 8690, pp. 834–849.

[34] J. Engel, J. Stückler, and D. Cremers, “Large-scale direct slam with stereo cameras,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 1935–1942.

[35] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[36] H. Strasdat, J. M. M. Montiel, and A. J. Davison, “Real-time monocular slam: Why filter?” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 2657–2664.

[37] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004.

[38] A. Davison, I. Reid, N. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1052–1067, June, 2007.

[39] A. J. Davison and N. Kita, “3d simultaneous localisation and map-building using active vision for a robot moving on undulating terrain,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–384–I–391 vol.1.

[40] W. Y. Jeong and K. M. Lee, “Visual slam with line and corner features,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 2570–2575.

[41] S. J. Julier and J. K. Uhlmann, “A counter example to the theory of simultaneous localization and map building,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 4, 2001, pp. 4238–4243 vol.4.

[42] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, “Consistency of the ekf-slam algorithm,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 3562–3568.

[43] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” *Proc. of 8th National Conference on Artificial Intelligence/14th Conference on Innovative Applications of Artificial Intelligence*, vol. 68, no. 2, pp. 593–598, 2002.

[44] E. Eade and T. Drummond, “Scalable monocular slam,” in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, ser. CVPR ’06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 469–476.

[45] R. Sim, P. Elinas, and M. Griffin, “Vision-based slam using the rao-blackwellised particle filter,” in *In IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 2005.

[46] D. Bagnell, G. Seyfarth, and Z. Batts, “Good, bad, and ugly of particle filters,” http://www.cs.cmu.edu/~16831-f12/notes/F14/16831_lecture05_gseyfarth_zbatts.pdf, 2014.

[47] J. Kim, K.-J. Yoon, and I. S. Kweon, “Bayesian filtering for keyframe-based visual slam,” *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 517–531, 2015.

[48] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *Proceedings - IEEE International Conference on Robotics and Automation*, no. June, 2011, pp. 3607–3613.

[49] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping using the Bayes tree,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[50] D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, June 2004, pp. I–652–I–659 Vol.1.

[51] D. Scaramuzza and F. Fraundorfer, “Visual odometry [tutorial],” *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.

[52] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard, “Simultaneous localization and mapping: Present, future, and the robust-perception age,” *CoRR*, vol. abs/1606.05830, 2016.

[53] C. Harris and M. Stephens, “A combined corner and edge detector,” in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[54] J. Shi and C. Tomasi, “Good features to track,” in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun 1994, pp. 593–600.

[55] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.

[56] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792.

[57] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[58] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: an efficient alternative to sift or surf,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.

[59] H. Li and R. Hartley, “Five-point motion estimation made easy,” *Proceedings - International Conference on Pattern Recognition*, vol. 1, pp. 630–633, 2006.

[60] L. Kneip, D. Scaramuzza, and R. Siegwart, “A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation,” in *CVPR 2011*, June 2011, pp. 2969–2976.

[61] D. Nister, “A minimal solution to the generalised 3-point pose problem,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, June 2004, pp. I–560–I–567 Vol.1.

[62] R. Horaud, B. Conio, O. Leboulleux, B. Lacolle, R. Horaud, B. Conio, O. Leboulleux, B. Lacolle, A. Analytic, R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle, “An Analytic Solution for the Perspective 4-Point Problem To cite this version : HAL Id : inria-00589990 An Analytic Solution for the Perspective 4Point Problem,” 2014.

[63] B. Li, L. Heng, G. H. Lee, and M. Pollefeys, “A 4-Point Algorithm for Relative Pose Estimation of a Calibrated Camera with a Known Relative Rotation Angle.”

[64] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An accurate O(n) solution to the PnP problem,” *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[65] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[66] B. Kitt, A. Geiger, and H. Lategahn, “Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme,” in *2010 IEEE Intelligent Vehicles Symposium*, June 2010, pp. 486–492.

[67] A. J. Davison, “3D Simultaneous Localisation and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain,” vol. 00, no. C, 2001.

[68] M. W. M. G. Dissanayake, P. Newman, S. Clark, and M. Csorba, “A Solution to the Simultaneous Localisation and Map Building (SLAM) Problem,” pp. 1–14, 2006.

[69] P. Pini, S. Member, and J. D. Tardós, “Independent Local Maps : Application to Monocular Vision,” vol. 24, no. 5, pp. 1–13, 2008.

[70] F. Dellaert and M. Kaess, “Square root sam: Simultaneous localization and mapping via square root information smoothing,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[71] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[72] F. Dellaert and M. Kaess, “Factor graphs for robot perception,” *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[73] S. Thrun and M. Montemerlo, “The graph SLAM algorithm with applications to large-scale mapping of urban structures,” in *International Journal of Robotics Research*, vol. 25, no. 5-6, 2006, pp. 403–429.

[74] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, Nov 2007, pp. 225–234.

[75] R. Mur-Artal and J. D. Tardos, “Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras,” *arXiv preprint arXiv:1610.06475*, 2016.

[76] D. Filliat, “A visual bag of words method for interactive qualitative localization and mapping,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 3921–3926.

[77] A. Angeli, D. Filliat, S. Doncieux, and J. A. Meyer, “Fast and incremental method for loop-closure detection using bags of visual words,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1027–1037, Oct 2008.

[78] D. Galvez-López and J. D. Tardos, “Bags of binary words for fast place recognition in image sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, Oct 2012.

[79] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular slam," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1188 – 1197, 2009, inside Data Association.

[80] C. R. Rao, "Advanced statistical methods in biometric research." 1952.

[81] H. F. Durrant-Whyte, "Uncertain geometry in robotics," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 23–31, Feb 1988.

[82] R. Smith, M. Self, and P. Cheeseman, "Estimating Uncertain Spatial Relationships in Robotics," *ArXiv e-prints*, Mar. 2013.

[83] J. Knuth and P. Barooah, "Error growth in position estimation from noisy relative pose measurements," *Robotics and Autonomous Systems*, vol. 61, no. 3, pp. 229 – 244, 2013.

[84] P. Druilhet and D. Pommeret, "Invariant conjugate analysis for exponential families," vol. 7, 12 2012.

[85] "Bumblebee2 stereo vision camera," https://www.ptgrey.com/bumblebee2-firewire-stereo-vision-camera-systems.

[86] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?"," in *Computer Vision — ECCV 2002*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 414–431.

[87] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, "Visual modeling with a hand-held camera," *International Journal of Computer Vision*, vol. 59, no. 3, pp. 207–232, Sep 2004.

[88] J. L. Schönberger and J. M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4104–4113.

[89] C. Geyer and K. Daniilidis, "Structure and motion from uncalibrated catadioptric views," *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, no. May, pp. I–279–I–286 vol.1, 2001.

[90] M. Pollefeys, R. Koch, and L. Van Gool, "Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Intrinsic Camera Parameters," 1998.

[91] A. Bhaskaran and A. Sridhar Rao, "Structure from Motion using Uncalibrated Cameras," pp. 1–4.

[92] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular slam," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, Oct 2008.

[93] P. Pinies and J. D. Tardos, "Large-scale slam building conditionally independent local maps: Application to monocular vision," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1094–1106, Oct 2008.

[94] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, 2010.

[95] D. Gamage and T. Drummond, "Reduced dimensionality extended kalman filter for slam in a relative formulation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 1365–1372.

[96] G. Klein and D. W. Murray, "Improving the agility of keyframe-based SLAM," in *Proc 10th European Conf on Computer Vision, Marseille, France*, 2008.

[97] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular slam in dynamic environments," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Oct 2013, pp. 209–218.

[98] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct 2015.

[99] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" *2010 IEEE International Conference on Robotics and Automation*, pp. 2657–2664, may 2010.

[100] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[101] M. Kaess, “Incremental Smoothing and Mapping,” *Work*, no. December, 2007.

[102] H. Strasdat, J. Montiel, and A. J. Davison, “Scale drift-aware large scale monocular slam.” in *Robotics: Science and Systems*, vol. 2, no. 3, 2010, p. 5.

[103] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, “Onboard imu and monocular vision based control for mavs in unknown in- and outdoor environments,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 3056–3063.

[104] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, “Fusion of imu and vision for absolute scale estimation in monocular slam,” *Journal of Intelligent and Robotic Systems*, vol. 61, no. 1-4, pp. 287–299, 2011.

[105] A. Tabibiazar and O. Basir, “Radio-visual signal fusion for localization in cellular networks,” in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*, Sept 2010, pp. 150–155.

[106] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[107] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *Antennas and Propagation, IEEE Transactions on*, vol. 34, no. 3, pp. 276–280, 1986.

[108] G. Wahba, “A least squares estimate of satellite attitude,” *SIAM Review*, vol. 7, no. 3, pp. 409–409, 1965. [Online]. Available: https://doi.org/10.1137/1007077

[109] K. W., “A solution for the best rotation to relate two sets of vectors,” *Acta Crystallographica Section A*, vol. 32, no. 5, pp. 922–923.

[110] F. Dellaert, “Factor graphs and gtsam: A hands-on introduction,” Georgia Institute of Technology, Tech. Rep., 2012.

[111] J. J. Moré, “The levenberg-marquardt algorithm: implementation and theory,” in *Numerical analysis*. Springer, 1978, pp. 105–116.

[112] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb 2008.

[113] N. Snavely, S. M. Seitz, and R. Szeliski, “Modeling the world from internet photo collections,” *Int. J. Comput. Vision*, vol. 80, no. 2, pp. 189–210, Nov. 2008.

[114] M. Bruckner, F. Bajramovic, and J. Denzler, “Geometric and probabilistic image dissimilarity measures for common field of view detection,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June, 2009, pp. 2052–2057.

[115] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: a survey,” *Found. Trends. Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280, Jul. 2008.

[116] J. Xiao, J. Zhang, J. Zhang, H. Zhang, and H. Hildre, “Fast plane detection for slam from noisy range images in both structured and unstructured environments,” in *Mechatronics and Automation (ICMA), 2011 International Conference on*, 2011, pp. 1768–1773.

[117] J. Zhou and B. Li, “Homography-based ground detection for a mobile robot platform using a single camera,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 4100–4105.

[118] D. E. Gustafson and W. C. Kessel, “Fuzzy clustering with a fuzzy covariance matrix,” in *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, vol. 17. IEEE, 1978, pp. 761–766.

[119] U. Kaymak and M. Setnes, “Fuzzy clustering with volume prototypes and adaptive cluster merging,” *Fuzzy Systems, IEEE Transactions on*, vol. 10, no. 6, pp. 705–712, 2002.

[120] Y. Mohan and S. G. Ponnambalam, “An extensive review of research in swarm robotics,” in *Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 140–145.

[121] A. Gautam and S. Mohan, “A review of research in multi-robot systems,” no. August, 2012, pp. 1–5.

[122] S. Saeedi, M. Trentini, M. Seto, and H. Li, “Multiple-robot simultaneous localization and mapping: A review,” *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[123] R. Egodagamage and M. Tuceryan, “A collaborative augmented reality framework based on distributed visual slam,” in *2017 International Conference on Cyberworlds (CW)*, Sept 2017, pp. 25–32.

[124] P. Robertson, M. G. Puyol, and M. Angermann, “Collaborative Pedestrian Mapping of Buildings Using Inertial Sensors and FootSLAM,” in *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR*, 2011, pp. 1366–1377.

[125] S. Saeedi, M. Trentini, and H. Li, “A hybrid approach for multiple-robot SLAM with particle filtering,” *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-December, pp. 3421–3426, 2015.

[126] V. Indelman, N. Michael, and F. Dellaert, “Incremental Distributed Robust Inference from Arbitrary Robot Poses via EM and Model Selection,” *RSS Workshop on Distributed Control and Estimation for Robotic Vehicle Networks*, pp. 8–11, 2014.

[127] V. Indelman, E. Nelson, J. Dong, N. Michael, and F. Dellaert, “Incremental distributed inference from arbitrary poses and unknown data association: Using collaborating robots to establish a common reference,” *IEEE Control Systems*, vol. 36, no. 2, pp. 41–74, April 2016.

[128] L. Paull, G. Huang, M. Seto, and J. J. Leonard, “Communication-constrained multi-AUV cooperative SLAM,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 509–516, 2015.

[129] L. Carlone, M. K. Ng, J. Du, B. Bona, and M. Indri, “Rao-blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 243–249, 2010.

[130] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, “A fast and accurate approximation for planar pose graph optimization,” *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 965–987, 2014.

[131] O. De Silva, G. K. I. Mann, and R. G. Gosine, “Development of a relative localization scheme for ground-aerial multi-robot systems,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 870–875, 2012.

[132] L. Carlone and R. Aragues, “A linear approximation for graph-based simultaneous localization and mapping,” *Robotics: Science and . . .* , p. 8, 2011.

[133] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, “A fast and accurate approximation for planar pose graph optimization,” *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 965–987, 2014.

[134] L. Carlone and A. Censi, “From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization,” *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 475–492, 2014.

[135] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, “Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 4597–4604, 2015.

[136] G. Calafiore, L. Carlone, and F. Dellaert, “Pose Graph Optimization in the Complex Domain: Lagrangian Duality, Conditions For Zero Duality Gap, and Optimal Solutions,” p. 15, 2015.

[137] D. Zou and P. Tan, “Coslam: Collaborative visual slam in dynamic environments,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 2, pp. 354–366, 2013.

[138] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, “On spectral clustering: Analysis and an algorithm,” *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[139] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.