Omnidirectional Vision for an Autonomous Surface Vehicle

Xiaojin Gong

Dissertation submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Christopher L. Wyatt, Chair
A. Lynn Abbott, Cochair
Daniel J. Stilwell
William T. Baumann
Naren Ramakrishnan

December 5, 2008
Blacksburg, Virginia

Keywords: Omnidirectional vision, calibration, range estimation, outlier rejection, motion detection, autonomous navigation
Acknowledgements

I owe a great debt of gratitude to many people who have helped me and made the research work in this dissertation possible.

First and foremost, I would like to thank my advisor, Dr. Chris Wyatt, for his guidance, inspiration, and encouragement over the years. His continuous support, considerable help, and patience have been of utmost importance for finishing my PhD study.

I am greatly indebted to my co-advisor, Dr. Lynn Abbott, who deserves my deepest gratitude for introducing me into the Computer Vision field, for providing many guidance and resources in the first two years of my PhD study. I am also greatly indebted to Dr. Daniel Stilwell, who has inspired me and helped me at all stages of this work.

I appreciate the efforts of the other two members of my PhD committee, Dr. William Baumann and Dr. Naren Ramakrishnan, for their time reviewing the dissertation and providing me with insightful comments.

I would like to recognize the contributions of my colleagues, in particular, Bin Xu, Anbumani Subramanian, Caleeb Reed, Darren Maczka, and Jamie Riggins. Without their support and assistance, the work would have been impossible or at least much less enjoyable to complete.

Furthermore, I would like to thank my family for their love and support. Thanks my parents for their instilling in me a love of learning and a desire to always do my best. It is their encouragement that keeps me going forward all along.

I thank all my friends for their support.

This research work was supported in part by the National Science Foundation via grant IIS-0238092, and by the Office of Naval Research via grants N00014-03-1-0444, N00014-05-1-0780, and N00014-05-1-0516.
Omnidirectional Vision for an Autonomous Surface Vehicle

Xiaojin Gong

Abstract

Due to the wide field of view, omnidirectional cameras have been extensively used in many applications, including surveillance and autonomous navigation. In order to implement a fully autonomous system, one of the essential problems is construction of an accurate, dynamic environment model. In Computer Vision this is called structure from stereo or motion (SFSM). The work in this dissertation addresses omnidirectional vision based SFSM for the navigation of an autonomous surface vehicle (ASV), and implements a vision system capable of locating stationary obstacles and detecting moving objects in real time.

The environments where the ASV navigates are complex and fully of noise, system performance hence is a primary concern. In this dissertation, we thoroughly investigate the performance of range estimation for our omnidirectional vision system, regarding to different omnidirectional stereo configurations and considering kinds of noise, for instance, disturbances in calibration, stereo configuration, and image processing. The result of performance analysis is very important for our applications, which not only impacts the ASV’s navigation, also guides the development of our omnidirectional stereo vision system.

Another big challenge is to deal with noisy image data attained from riverine environments. In our vision system, a four-step image processing procedure is designed: feature detection, feature tracking, motion detection, and outlier rejection. The choice of point-wise features and outlier rejection based method makes motion detection and stationary obstacle detection efficient. Long run outdoor experiments are conducted in real time and show the effectiveness of the system.
Contents

Abstract ................................................................. iii
List of Figures ......................................................... vii
List of Tables .......................................................... xi

1 Introduction 1
  1.1 Motivation ......................................................... 1
  1.2 Survey of Vision-based Autonomous Navigation ............... 2
  1.3 Contributions ...................................................... 4
  1.4 Organization ....................................................... 5

2 Omnidirectional Camera 6
  2.1 Introduction to Omnidirectional Camera ......................... 6
    2.1.1 Terminology .................................................. 6
    2.1.2 Classification ................................................. 7
    2.1.3 Catadioptric Cameras ....................................... 8
  2.2 Introduction to Geometries of Catadioptric Cameras ........... 9
    2.2.1 Single View Geometry ...................................... 9
    2.2.2 Two View Geometry ......................................... 13
    2.2.3 Epipolar Geometry .......................................... 18
  2.3 Summary .......................................................... 22

3 Omnidirectional Camera Calibration 23
  3.1 Related Work ..................................................... 24
5.3 Outlier Rejection ............................................... 71
  5.3.1 Matching Residual Based Outlier Rejection ............ 71
  5.3.2 N-ocular Stereo Based Outlier Rejection .............. 75
  5.3.3 Experimental Results .................................... 76
5.4 Motion Detection .............................................. 80
  5.4.1 Related Work ............................................ 80
  5.4.2 Challenges in Our Application .......................... 81
  5.4.3 A Two-Step Motion Detection Algorithm ............... 82
  5.4.4 Experimental Results .................................... 83
5.5 Summary ...................................................... 87

6 Experiments and Validation 88
  6.1 Testbed Setup .............................................. 88
  6.2 Omnivision System ......................................... 89
  6.3 Experimental Results ...................................... 92

7 Conclusions and Future Work 96
  7.1 Conclusions ............................................... 96
  7.2 Future Work ............................................... 97
Bibliography ..................................................... 98
List of Figures

2.1 A paracatadioptric camera. ............................................. 10
2.2 The paracatadioptric projection model. ............................... 10
2.3 A corridor image taken by the paracatadioptric camera. .......... 11
2.4 An image taken by the paracatadioptric camera at Claytor Lake. . 11
2.5 A hypercatadioptric camera. ............................................ 12
2.6 The hypercatadioptric projection model. ............................... 12
2.7 Stereo geometry .......................................................... 14
2.8 The intersection of two lines. .......................................... 16
2.9 Epipolar geometry in an omnistereo system. .......................... 19
2.10 A paraomnidirectional image marked with feature points. ........ 20
2.11 A parao-image with correspondences and epipolar lines. ........ 20

3.1 The general projection model for an omnidirectional camera. ...... 25
3.2 An image used in calibration. ........................................... 28
3.3 The simulated planes in calibration. ................................... 32
3.4 Back-projection error and the standard deviation vs. the noise level in the checkerboard. ..................................................... 34
3.5 Relative error of $\alpha, \beta$, the mean values and standard deviations vs. the noise level in the checkerboard. .......................... 34
3.6 Absolute error of $u_0, v_0$, the mean values and standard deviations vs. the noise level in the checkerboard. .......................... 34
3.7 Back-projection error and the standard deviation vs. the noise level in corner detection. ..................................................... 35
3.8 Relative error of $\alpha, \beta$, the mean values and standard deviations vs. the noise level in corner detection. ........................................ 35
3.9 Absolute error of $u_0, v_0$, the mean values and standard deviations vs. the noise level in corner detection. ............................... 35
3.10 Back-projection error and the standard deviation vs. the number of planes. ................................................................. 36
3.11 Relative error of $\alpha, \beta$, the mean values and standard deviations vs. the number of planes. ........................................ 36
3.12 Absolute error of $u_0, v_0$, the mean values and standard deviations vs. the number of planes. ........................................ 36
3.13 Back-projection error and the standard deviation using two groups of image planes. ......................................................... 37
3.14 Relative error of $\alpha, \beta$, the mean values and standard deviations using two groups of image planes. ........................................ 37
3.15 Absolute error of $u_0, v_0$, the mean values and standard deviations using two groups of image planes. ........................................ 37
3.16 An image taken by the first hypercatadioptric camera. .......... 40
3.17 An image taken by the second hypercatadioptric camera. ......... 40
3.18 3D reconstruction results. ................................................. 40

4.1 A disparity distribution for case 1. ........................................ 47
4.2 A disparity distribution for case 2. ........................................ 47
4.3 The distribution of $\frac{|\Delta P_{\text{r}}|}{|P|}$ for a single moving paracatadioptric camera. 52
4.4 The distribution of $\frac{|\Delta P_{\text{r}}|}{|P|}$ for two fixed hypercatadioptric cameras. 52
4.5 A reconstruction with a translation error. .............................. 54
4.6 The distribution of $\frac{|\Delta P_{\text{r}}|}{|P|}$ for a single moving paracatadioptric camera. 55
4.7 The distribution of $\frac{|\Delta P_{\text{r}}|}{|P|}$ for two fixed hypercatadioptric cameras. 55
4.8 The distribution of $\frac{|\Delta P_{\text{r}}|}{|P|}$ for a single moving paracatadioptric camera. 57
4.9 The distribution of $\frac{|\Delta P_{\text{r}}|}{|P|}$ for two fixed hypercatadioptric cameras. 57
4.10 The distribution of $\frac{|\Delta P_{\text{r}}|}{|P|}$ for a single moving hypercatadioptric camera. ......................................................... 60

viii
4.11 The distribution of $|\Delta P|/|P|$ for a single moving hypercatadioptric camera. ................................................. 60
4.12 The distribution of $|\Delta P_A|/|P|$ for a single moving hypercatadioptric camera. ................................................. 61
4.13 The distribution of $|\Delta P|/|P|$ for a single moving hypercatadioptric camera. ................................................. 61
4.14 The distribution of $|\Delta P_K|/|P|$ for a single moving hypercatadioptric camera. ................................................. 62
4.15 The distribution of $|\Delta P|/|P|$ for a single moving hypercatadioptric camera. ................................................. 62
4.16 The distribution of $|\Delta P_A|/|P|$ for a single moving hypercatadioptric camera. ................................................. 63
4.17 The distribution of $|\Delta P_K|/|P|$ for a single moving hypercatadioptric camera. ................................................. 63
4.18 The distribution of $|\Delta P_K|/|P|$ for two fixed hypercatadioptric cameras. ................................................. 64
4.19 The distribution of $|\Delta P_P|/|P|$ for two fixed hypercatadioptric cameras. ................................................. 64
4.20 The distribution of $|\Delta P_A|/|P|$ for two fixed hypercatadioptric cameras. ................................................. 65
4.21 The distribution of $|\Delta P_P|/|P|$ for two fixed hypercatadioptric cameras. ................................................. 65

5.1 Omnidirectional images with features marked as red points. ........ 70
5.2 The trinocular geometry. ................................................................. 76
5.3 Experimental results of two-step outlier rejection. ...................... 78
5.4 Experimental results of two-step outlier rejection shown on the topomap. 79
5.5 An omnidirectional image marked with features and optical flows. 82
5.6 Experimental results of motion detection. ................................. 85
5.7 Experimental results of motion detection. ................................. 86
5.8 An overview of vision processing. ............................................... 87

6.1 A photo of the ASV, equipped with a paracatadioptric camera. .... 89
6.2 A photo of the ASV, equipped with two hypercatadioptric cameras. 89
6.3 Overview of the omnivision system. .......................................... 91
6.4 A result of online range estimation using a paracatadioptric camera. 93
6.5 A result of online range estimation using a paracatadioptric camera. 94
6.6 A result of online range estimation using two hypercatadioptric cameras. . . . . . . 95

Unless otherwise noted, all images are property of the author.
# List of Tables

2.1  Unifying Model Parameters ........................................ 14  
3.1  Simulated Data .................................................. 32  
3.2  Calibration Results ............................................. 39  
5.1  Feature Detection Parameters .................................... 68  
5.2  Feature Tracking Parameters ..................................... 71  
5.3  Results of Outlier Rejection .................................... 77
Chapter 1

Introduction

1.1 Motivation

This work is motivated by the navigation of an autonomous surface vehicle (ASV) in a riverine environment [58]. In the pursuit of a truly autonomous vehicle [14], it is necessary to synthesize many techniques, for instance, control theory, artificial intelligence, and computer vision if a visual sensor is used. Typically, a robot perceives the outside world using visual, auditory or some other type of sensors, models the surrounding environment, designs an appropriate plan to execute a given task, as well as handles unexpected events. This dissertation focuses on the omnidirectional vision based environment modeling component.

The ASV, being studied here, is a product of the Autonomous Systems and Controls Laboratory (ASCL), equipped with a global positioning system (GPS) and a gyroscope to acquire position (longitude, and latitude) and attitude (pitch, roll, and yaw) information. In addition, a camera is adopted as a range finding sensor to estimate the distance to surrounding obstacles. The reason for choosing a camera is that, in contrast to active range finding sensors such as laser or radar, a camera is a passive sensor which has lower power consumption. Moreover, a camera captures a richer set of features, from which obstacles can be detected and the map of environment reconstructed, so that the robot is capable of navigating autonomously in a large variety of environments.

For a conventional pinhole camera, the primary drawback is the limitation on
the field of view. This is not a constraint when omnidirectional cameras are used. An omnidirectional camera provides a 360° horizontal field of view so that it can capture images of the entire surrounding areas. This is an attractive characteristic to autonomous navigation because the robot is therefore able to see obstacles appearing in any horizontal direction. In addition, as pointed out by T. Svoboda [69], more information in an image also contributes to stable computation for 3D reconstruction and motion detection. The overlap in omnidirectional vision is almost complete, implying that most correspondences will not be lost when the robot is moving. Regarding all these advantages, omnidirectional cameras are hence adopted.

Generally speaking, this work is concerned with the use of omnidirectional vision in the ASV’s navigation. It addresses the riverine environment modeling problem using omnidirectional cameras, and implements an omnidirectional vision system capable of detecting obstacles in real-time.

1.2 Survey of Vision-based Autonomous Navigation

This dissertation covers subjects from omnidirectional camera geometry, to omnidirectional camera calibration, performance analysis, outlier rejection, and motion detection. I am not going to discuss related work on each topic here, rather I introduce several vision-based autonomous navigation frameworks.

- **Vision-based autonomous air vehicle.** In [33], a real-time 3D vision system for autonomous small air vehicles, developed by Carnegie Mellon University, is presented. It utilizes a conventional video camera to capture ground images. Feature selection, tracking and structure from motion are three sub modules which make up the vision processing system. An extended KLT coping with the affine transformation, illumination change, and large motion is designed to select and track features. The relative displacement of the camera at two time instants is calculated by estimating the Essential matrix [11] based on the features matched in two image frames. In [36] and
[78], a vision-based control system for micro air vehicles is presented which is developed by University of Florida. It takes the results of feature tracking as inputs to the extended Kalman Filter (EKF) to estimate the state of vehicle.

- **Vision-based unmanned ground vehicle.** In[7], [2], [48], [49], [10], and [81], omnidirectional vision systems for the navigation of unmanned ground vehicles are tested in indoor and structured environments. In [10], [81], and [48], the designs of omnidirectional cameras and the representations of environments are emphasized. Assuming the ground is a horizontal plane, they use inverse projection and homography geometry [55] to detect obstacles and recover their 3D positions. In [7], omnidirectional images are transformed into panoramic ones and dense depth maps constructed. [2] employs an omnidirectional camera together with a pinhole camera as a hybrid stereo system. The correspondences in two different images are matched by Inverse Perspective Transform (IPT) with respect to the ground plane.

- **Autonomous surface vehicle.** There are several ASVs developed independently in research labs and companies, for instance, the ACES at MIT [44] and the WASP at Santa Clara University [43]. The most common use of ASVs is as a communication and navigation aid for AUV navigation, hence most work in this area places emphasis on the ASV’s dynamic and mechanical design and its communication capabilities. To our knowledge, [65] is the only work published that studies vision-based autonomous navigation for ASVs. It uses six pinhole cameras to obtain the 360° field of view. Calibration is then necessary to align the images taken by different cameras. Harris corner detection is used to select features and normalized correlation is applied to track features in successive frames. With the assumption that the water surface is horizontal, they simplify a camera to an optical sonar model and therefore estimate the range from a single image. They also use multi-camera stereo to estimate range by calculating the Essential matrix [11]. A feature clustering algorithm based on a fixed clustering window is designed to make tracking more robust and to detect moving features.
In summary, vision-based autonomous navigation has been receiving considerable attention, employing either perspective or omnidirectional cameras. However, most of the existing work is applied to ground (RoboCup robots, exploration vehicles, etc.) and airborne (unmanned aerial vehicles, UAVs) vehicles. There is only one published work [65] tackling vision-based navigation for ASVs, other than our own [26], [27].

1.3 Contributions

To our knowledge, the work in this dissertation is the first applying omnidirectional cameras in the navigation of ASVs. One of the essential differences between this work and that mentioned above lies on the background environment. In many applications, the ground, or even the water surface [65], is taken as a reference plane. The background’s planarity is assumed. However, in our application, due to the presence of waves and specular reflections, it is inappropriate to view the water surface as a plane. The large-scale complex environments where ASVs are operated, together with the use of omnidirectional vision system, complicate this work and also distinguish our work.

The contributions of this dissertation to the omnidirectional vision based autonomous navigation are as follows. This work

- investigated the performance of the omnidirectional vision system, which includes the maximum detectable range and the sensitivity of range estimation with respect to noise existing in the system (refer to Chapter 4);

- implemented a calibration method to estimate intrinsic parameters of omnidirectional cameras, and studied the accuracy of calibration (refer to Chapter 3);

- designed an outlier rejection method to detect both stationary and moving obstacles (refer to Chapter 5);

- developed an omnidirectional vision system, which is able to detect obstacles in real time (refer to Chapter 6).
Two important conclusions can be drawn from this work:

- Omnidirectional stereo is sensitive to differences in camera miscalibration.
- Omnidirectional stereo is sensitive to noise for points near the baseline direction.

1.4 Organization

The rest of this dissertation is organized as follows. Chapter 2 introduces the background of omnidirectional vision, including single view geometry, two view geometry and the epipolar constraint. Chapter 3 presents a method of calibrating paracatadioptric and hypercatadioptric cameras, and numerically investigates the calibration performance using Monte Carlo simulation. Chapter 4 investigates the range estimation performance of the omnidirectional stereo vision system, with respect to calibration noise, stereo configuration, and image processing. Chapter 5 introduces the omnidirectional image processing, from feature detection and feature tracking to outlier rejection based obstacle detection methods. Chapter 6 introduces the framework of the entire vision system and presents experimental results. Chapter 7 draws conclusions and points out directions of future work.
Chapter 2

Omnidirectional Camera

Since omnidirectional cameras are the foundation on which the entire navigation system relies, it is necessary to introduce the background of omnidirectional cameras and derive the geometries of single and two view projections for both parabolic catadioptric (paracatadioptric) cameras and hyperbolic catadioptric (hypercatadioptric) cameras. Their epipolar geometries are also presented at the end of the chapter.

2.1 Introduction to Omnidirectional Camera

2.1.1 Terminology

- *Omnidirectional* is a more technical term, used synonymously with the term *panoramic* [6] in the community. It refers not to a view captured in each direction, but to one captured in all horizontal directions while limited in the vertical direction. Typically, an omnidirectional camera produces a 360° field of view horizontally and 80° to 120° view vertically. The vertical field of view depends on the design of the camera.

- *Catadioptric* is a term first used by Nayar et al. [50] in 1997. It pertains to both *dioptic* and *catoptic* elements. A *dioptic* element is a refractive element, for example, a lens, and a *catoptic* element is a reflection element, like a mirror. A catadioptric camera system obtains a wide field of view by
combining a camera with a mirror. This type of camera is currently the most popular for omnidirectional vision.

- **Single View Point (SVP)** means the unique center of projection [53]. An example is the pinhole of a perspective camera, for which the scene is observed from a single point. Scenes can also be captured from a locus of viewpoints, called a *caustics* [71]. Omnidirectional sensors are classified into SVP (or central cameras) and non-SVP (or non-central cameras) categories. SVP is a preferable characteristic for an omnidirectional camera. An SVP camera follows the perspective projection laws, while a non-SVP camera does not respect the laws, leading to complicated mapping functions.

- **Uniform Resolution** means that the resolution is the same everywhere in an image. A variety of omnidirectional cameras do not have uniform resolution due to the use of conic mirrors, with higher resolution near the center and lower resolution toward the boundary.

### 2.1.2 Classification

As mentioned above, SVP / non-SVP is a criterion to classify the omnidirectional cameras. In the literature, the more common way to categorize these sensors is according to the techniques and fabrication adopted, by which, omnidirectional cameras are classified into the following categories [85] [53]:

- **Fisheye lens imaging systems.** This type of cameras uses a special fisheye lens rather than a conventional camera lens to extend the field of view. A fisheye lens has a very short focal length that enables a camera to acquire a hemispherical wide-angle view. The disadvantages for this kind of system are: 1) The resolution of acquired images is non-uniform. Images have good resolution in center region but poor resolution in the marginal region. 2) A fisheye lens produces a severe radial distortion, which is difficult to remove. 3) A fisheye lens has a locus of viewpoints. The non-SVP property makes it hard to generate a complete perspective image. Despite these drawbacks, the wide field of view and the low cost make a fisheye lens attractive. Several
researchers worked on distortion compensation [5] and mapping functions [62] for this type of sensors.

- **Multiple image acquisition systems.** The field of view in this kind of system is extended by combining multiple images captured in different directions using a mosaic technique. There are two ways to obtain such images. One is to rotate a camera around a vertical axis and capture images at different time instants [54], and the other is to use multiple cameras that are precisely assembled within a fixed structure [5] [12]. The obtained images have high and uniform resolution. However, it takes time to register the set of images. For rotating cameras, in particular, it requires the system to use moving parts, which is time consuming and not preferred in robotics. It also restricts its use to static scenes and non-real-time applications. For multiple cameras, the alignment and calibration among each camera are difficult.

- **Catadioptric imaging systems.** A catadioptric instrument is an optical system combining a reflective element with a refractive element, typically a mirror with a lens. The mirror is placed at the front of the camera. What the camera captures is the scene on the mirror, enhancing the field of view dramatically. The selection of the shape, position, and orientation of the mirror is complicated because of the desire to keep the effective viewpoint of the system fixed. Catadioptric systems can be classified into central and non-central catadioptric cameras according to the uniqueness of an effective viewpoint. The central catadioptric cameras are much more popular than the others due to the simplicity of projection.

### 2.1.3 Catadioptric Cameras

As stated earlier, a catadioptric camera combines a mirror and a lens in order to produce images having an omnidirectional field of view. With the advancement of technology, catadioptric cameras have become popular within omnidirectional vision and applied extensively in visualization and navigation.

The first patented omnidirectional catadioptric camera was invented by D.W. Rees in 1970 [57]. This camera combined a hyperboloid mirror with a conventional
perspective camera to increase the field of view for a weapon operator seated in a turret-like device. Yagi et al. [85] introduced the catadioptric sensor into robotics twenty years later. Since then, catadioptric cameras have been studied extensively. Nayar et al. [50] [23] [4] [22] [51] designed a number of catadioptric cameras based on a thorough analysis of the optical properties of conic mirrors. Geyer et al. [17] [18] [16] [20] unified the projection models of different catadioptric cameras and proposed a rectification method to address the registration problem between two distorted omnidirectional images and tried to solve the structure from motion problem for these type of camera. The omnistereo geometry was also studied by Svoboda et al. [69].

The mirror used in a catadioptric camera for reflection can be a planar mirror, or a mirror with a conic profile, such as a parabolic or hyperbolic reflecting surface. The most important subclass of all catadioptric cameras is the one which has a single viewpoint. Parabolic and hyperbolic mirrors have been shown to produce a single viewpoint when the camera and the mirror are aligned precisely. In our work, we utilize both parabolic and hyperbolic catadioptric cameras for the navigation of the ASV.

### 2.2 Introduction to Geometries of Catadioptric Cameras

In this subsection, I introduce the image formation and stereo geometries for both paracatadioptric and hypercatadioptric cameras.

#### 2.2.1 Single View Geometry

There are two sequential projections when a central catadioptric camera forms an image. A 3D scene point is first projected onto the reflecting surface, and then projected either orthographically or perspectively onto the image plane. The second projection depends on the conic profile of the mirror. For instance, it is an orthographic projection when the mirror is a paraboloid and a perspective projection for a hyperbolic mirror.
Paracatadioptric Camera

A paracatadioptric camera system is composed of a parabolic mirror and an orthographic lens camera. Figure 2.1 is a picture of the camera we use. The mirror is placed on the top of a clear tube and the camera in a waterproofing case. An important property of the parabolic reflecting surface is that the reflection of a ray directed along the line through the focus $F$ parallels the axis $Z$, as shown in Figure 2.2. Therefore, the unique effective viewpoint is obtained when the camera is placed at the infinite point, which is implemented using an orthographic lens. The camera and the mirror must be properly aligned with respect to each other.

Figure 2.2: The paracatadioptric projection model.

Figure 2.1: A paracatadioptric camera.

Figure 2.2 delineates the geometry of the paracatadioptric projection. A scene point $P = [X, Y, Z]^T$ in a three dimensional space is first projected onto the parabolic mirror at the point $p = [x, y, z]^T$, and then projected orthographically onto the image plane at the point $p' = [x', y']^T$. Due to the orthographic projection, the coordinate values of the 2D image point $p'$ are equivalent to the values of $p$ for $x$ and $y$ coordinates, that is, $p' = [x', y']^T = [x, y]^T$.

By fixing the origin of the coordinate system at the focus, the equation of the
A parabolic mirror is defined as
\[
\sqrt{x^2 + y^2 + z^2} = -z + 2h,
\] (2.1)
where, \(h\) is the distance from the vertex to the focus.

Since \(p\) is on the vector \(\overrightarrow{OP}\), we have
\[
p = \lambda \mathbf{P},
\] (2.2)
where \(\lambda\) is a scalar. Substituting Equation (2.2) into the parabolic equation, we obtain
\[
\lambda = \frac{2h}{Z + |\mathbf{P}|},
\] (2.3)
where \(|\mathbf{P}|\) is \(L_2\) norm of the vector \(\mathbf{P}\), \(|\mathbf{P}| = \sqrt{X^2 + Y^2 + Z^2}\).

To get an impression of what the captured images look like, I present two example images in Figure 2.3 and 2.4 which were taken by our paracatadioptric camera in the corridor in Whittemore Hall and at Claytor Lake respectively.

![Figure 2.3: A corridor image taken by the paracatadioptric camera.](image)

![Figure 2.4: An image taken by the paracatadioptric camera at Claytor Lake.](image)

**Hypercatadioptric Camera**

A hypercatadioptric camera system is composed of a hyperbolic mirror and a perspective camera. Figure 2.5 is the hypercatadioptric camera we use. The
reflection property of a hyperboloid is that a ray lying along the line through the focus \( F_1 \) reflects in such a way that the outgoing ray passes through the second focus \( F_2 \). Referred to this property, the unique effective viewpoint is obtained when the perspective camera is placed at the second focus and aligned properly. The projection geometry is shown in Figure 2.6.

By fixing the origin of the coordinate system at \( F_1 \), one of the hyperbola’s foci, the equation of the hyperbolic reflecting surface is defined as

\[
\frac{(z - c)^2}{a^2} - \frac{x^2 + y^2}{b^2} = 1,
\]

where, \( c = \sqrt{a^2 + b^2} \), \( a \) is the semimajor axis, and \( b \) is the semiminor axis of the hyperbola.

A scene point \( \mathbf{P} = [X, Y, Z]^T \) is projected onto the hyperbolic mirror at point \( \mathbf{p} = [x, y, z]^T \), and subsequently projected onto the image plane at point \( \mathbf{p}' = [x', y']^T \). We have

\[
\mathbf{p} = \lambda \mathbf{P},
\]

where

\[
\lambda = \frac{b^2}{cZ + a|\mathbf{P}|},
\]
and
\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \frac{-f}{-z + 2c} \begin{bmatrix} x \\ y \end{bmatrix},
\] (2.7)
where \( f \) is the focal length of the perspective camera.

Combining equation (2.4), (2.5), and (2.7), we get
\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \frac{-b^2 f}{(a^2 + c^2)Z + 2ac|P|} \begin{bmatrix} X \\ Y \end{bmatrix}. \tag{2.8}
\]

A Unifying Projection Model

We here unify the paracatadioptric and hypercatadioptric projection models, following Toepfer et al. [72]. Although Geyer [18] provided a more general unifying model, fitting all central catadioptric systems, showing that all projection models are isomorphic to projective mappings from a sphere to a plane with a projection center on the diameter perpendicular to the plane, the unification we present is a straightforward modification only for the type of cameras we use in our ASV.

Considering the hypercatadioptric projection model presented in Equation (2.8), if we divide \( 2ac \) in both the numerator and the denominator, we obtain
\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \frac{-f \cdot \frac{1}{2}(\epsilon - 1/\epsilon)}{\frac{1}{2}(\epsilon + 1/\epsilon)Z + |P|} \begin{bmatrix} X \\ Y \end{bmatrix}, \tag{2.9}
\]
where \( \epsilon = c/a \) is the eccentricity of the hyperbola.

Comparing both paracatadioptric and hypercatadioptric projection models, the following formula is obtained. The parameters listed in Table 2.1 convert equation (2.10) to either a paracatadioptric or hypercatadioptric model.
\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \frac{f \xi}{\zeta Z + |P|} \begin{bmatrix} X \\ Y \end{bmatrix}. \tag{2.10}
\]

2.2.2 Two View Geometry

In order to recover the 3D location for a scene point, a pair of images is required. This is well known as the stereopsis problem in computer vision. Stereopsis aims
<table>
<thead>
<tr>
<th>Camera</th>
<th>Parameter</th>
<th>$\xi$</th>
<th>$\zeta$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paracatadioptric camera</td>
<td></td>
<td>2$h$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Hypercatadioptric camera</td>
<td></td>
<td>$-\frac{1}{2}(\epsilon - 1/\epsilon)$</td>
<td>$\frac{1}{2}(\epsilon + 1/\epsilon)$</td>
<td>the focal length</td>
</tr>
</tbody>
</table>

Table 2.1: Unifying Model Parameters

to locate the 3D position by its given correspondences in two images. These two images might be captured simultaneously by two cameras in a fixed stereo system, or captured sequentially by a moving camera.

Let $(X_1, Y_1, Z_1)$ and $(X_2, Y_2, Z_2)$ be two coordinate systems, the former fixed in the first camera system and the latter in the second one such that their origins coincide with the foci of the respective conic mirrors, as shown in Figure 2.7. The translation vector from the first coordinate system to the second one, also called the baseline, is $\mathbf{T} = [t_x, t_y, t_z]^T$.

![Figure 2.7: Stereo geometry](image)

Given a point $\mathbf{p}_1 = [x_1', y_1']^T$ in the first image, its correspondence $\mathbf{p}_2' = [x_2', y_2']^T$ can be located in the second image by a feature matching algorithm. Assume that
$p_1 = [x_1, y_1, z_1]^T$ and $p_2 = [x_2, y_2, z_2]^T$ are the points on two conic mirrors that are back projected from $p'_1$ and $p'_2$, respectively. With $p_1$ and $p_2$, the scene point $P = [X, Y, Z]^T$ is uniquely determined in the 3D space by finding the intersection of two coplanar lines $O_1p_1$ and $O_2p_2$ in the ideal stereo model without noise. Any real stereo system suffers from noise due to the errors introduced in correspondence matching and stereo configuration, resulting in a skewness, that is, the lines $O_1p_1$ and $O_2p_2$ are not in the same plane, as shown by the blue points and lines in Figure 2.7. We hereby select the point $P_{2e}$ on the line $O_1p_1$ as the recovered 3D point. It makes more sense to choose this point than the midpoint of common perpendicular to these two rays, because we assume most of the noise is from correspondence matching and the stereo configuration. In addition, the concept of midpoint is meaningless under projective distortion [30].

Before calculating the intersection in terms of these two rays, I introduce a theorem describing how to find the intersection of any two lines in a three dimensional space, following which we obtain a formulation of the line intersection.

**Theorem 1.** [31] Let $x_1$, $x_2$, $x_3$ and $x_4$ be four noncolinear points in a three dimensional space, then the intersection point $x$ of line $x_1x_2$ and line $x_3x_4$ can be found by

$$x = x_1 + (x_2 - x_1) \left( \frac{c \times b}{|a \times b|^2} \cdot (a \times b) \right) \quad (a \times b \neq 0) \quad (2.11)$$

where

$$a = x_2 - x_1 \quad b = x_4 - x_3 \quad c = x_3 - x_1 \quad (2.12)$$

When the four points are coplanar, the line intersection is the true intersection; otherwise, the intersection found by Theorem 1 is the one on the line $x_1x_2$, as the point $x$ shown in Figure 2.8(1).

**Derivation of the 3D Localization Formula**

Referring to Theorem 1, we derive the formula which explicitly locates the 3D point in an omnidirectional stereopsis as follows:

**Step 1:** Transform $p_2$ and $O_2$ from the second coordinate system to the first. We take the first coordinate system as the reference one. In order to
calculate the intersection of $\overrightarrow{O_1p_1}$ and $\overrightarrow{O_2p_2}$, we need to transform the coordinates of $p_2$ from the second coordinate system to the first one by

$$p_{2}^{1st} = Rp_2 + T, \quad (2.13)$$

where $p_{2}^{1st} = [x_2^{1st}, y_2^{1st}, z_2^{1st}]^T$ is the coordinates of $p_2$ in the first coordinate system and $R$ is the rotation matrix.

Let $\alpha$, $\beta$, and $\theta$ be the Euler angles that the second coordinate system rotates about axis $X_1$, $Y_1$ and $Z_1$ respectively in the counterclockwise direction. Then $R$ is

$$R = R_z R_y R_x, \quad (2.14)$$

where,

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, \quad R_y = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}, \quad R_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

Following the transformation (2.13), we get $O_2$’s coordinates in the first coordinate system as

$$O_{2}^{1st} = T. \quad (2.16)$$

**Step 2: Calculate the intersection in the first coordinate system.** Now we are able to substitute $x_1 = O_1$, $x_2 = p_1$, $x_3 = O_{2}^{1st}$, and $x_4 = p_{2}^{1st}$ into Theorem 1, and get

$$P = p_1 F, \quad (2.17)$$
where,

\[ \mathcal{F} = \frac{(c \times b) \cdot (a \times b)}{|a \times b|^2}, \quad (a \times b \neq 0), \]  

(2.18)

and

\[
\begin{align*}
  a &= x_2 - x_1 = p_1 - O_1 = p_1 \\
  b &= x_4 - x_3 = p_{21} - O_{21} = R p_2 \\
  c &= x_3 - x_1 = O_{21} - O_1 = T.
\end{align*}
\]  

(2.19)

\[ \square \]

**Remark 2.2.1.** The formula of 3D localization degenerates when \(a \times b = 0\), which implies that \(O_1 p_1\) and \(O_2 p_2\) cannot be colinear.

**Remark 2.2.2.** The formula of 3D localization is suitable to any omnistereo configuration.

**Remark 2.2.3.** The formula of 3D localization is suitable to an arbitrary omnidirectional system. The only change needed is to substitute the projection model by which the points \(p_i\) on the mirrors are projected into the image points \(p'_i\) \((i = 1 \text{ or } 2)\), which is specific to different types of omnidirectional cameras. For instance,

1. For a paracatadioptric camera, the relation between \(p_i\) and \(p'_i\) follows

\[
\begin{bmatrix}
  x_i \\
  y_i \\
  z_i
\end{bmatrix} =
\begin{bmatrix}
  x'_i \\
  y'_i \\
  \frac{4h^2-(x'^2+y'^2)}{4h}
\end{bmatrix} =
\begin{bmatrix}
  x'_i \\
  y'_i \\
  \frac{\epsilon^2-(x'^2+y'^2)}{2}\epsilon
\end{bmatrix}.
\]  

(2.20)

2. For a hypercatadioptric camera, the relation between \(p_i\) and \(p'_i\) follows

\[
\begin{bmatrix}
  x_i \\
  y_i \\
  z_i
\end{bmatrix} =
\begin{bmatrix}
  0 \\
  0 \\
  2c
\end{bmatrix} +
\begin{bmatrix}
  x'_i \\
  y'_i \\
  f
\end{bmatrix} \frac{-b^2}{c f - a \sqrt{x'^2 + y'^2 + f^2}}.
\]  

(2.21)

Replacing the parameters \(a, b, c\) with those in the unifying model, we get the new formula as

\[
\begin{bmatrix}
  x_i \\
  y_i \\
  z_i
\end{bmatrix} = a \left\{ \begin{bmatrix}
  0 \\
  0 \\
  2\epsilon
\end{bmatrix} +
\begin{bmatrix}
  x'_i \\
  y'_i \\
  f
\end{bmatrix} \frac{1 - \epsilon^2}{\epsilon f - \sqrt{x'^2 + y'^2 + f^2}} \right\},
\]

(2.22)
where $\epsilon = \xi + \zeta$. In this equation, the parameter $a$ can be dropped when calculating the intersection of two vectors.

### 2.2.3 Epipolar Geometry

*Epipolar geometry* [11] describes a constraint on pairwise correspondences between images. Given a point on one image, its corresponding point on the other image is confined to lie on a line which is called the *epipolar line*. The epipolar constraint is an attractive constraint for correspondence matching algorithms, especially for wide baseline stereo matching, as the correspondence searching space is reduced rapidly, from two to one dimension. On the other hand, the epipolar constraint can also be recovered from a set of correspondences and hence be used to estimate the translation and rotation between two cameras [42].

In this subsection, I first demonstrate the epipolar constraint of an omnistereo system, and then introduce a method to estimate the *essential matrix* [11], which is an algebraic representation of the epipolar constraint.

**Epipolar Constraint**

Let $\mathbf{p}_1^\prime = [x_1^\prime, y_1^\prime]^T$ and $\mathbf{p}_2^\prime = [x_2^\prime, y_2^\prime]^T$ be pairwise correspondences in two images. Given $\mathbf{p}_1^\prime$ in the first image and its projection point $\mathbf{p}_1 = [x_1, y_1, z_1]^T$ on the mirror, $\mathbf{p}_2^\prime$’s back-projected point $\mathbf{p}_2 = [x_2, y_2, z_2]^T$ must lie on the line $\mathbf{p}_2 \mathbf{e}_2$ (as the red curve shown in Figure 2.9), which is the intersection of the plane $\mathbf{p}_1 \mathbf{O}_1 \mathbf{O}_2$ (called the *epipolar plane*) with the second mirror surface. Hence, $\mathbf{p}_2^\prime$ is constrained to the projected line of $\mathbf{p}_2 \mathbf{e}_2$ in the second image plane.

As the vectors $\mathbf{O}_1 \mathbf{p}_1$, $\mathbf{O}_2 \mathbf{p}_2$, and $\mathbf{O}_1 \mathbf{O}_2$ are coplanar, the two correspondences $\mathbf{p}_1$ and $\mathbf{p}_2$ satisfy the following constraint

$$\mathbf{p}_1^T \cdot (\mathbf{T} \times \mathbf{Rp}_2) = 0 \quad \implies \quad \mathbf{p}_1^T \mathbf{Ep}_2 = 0,$$

where $\mathbf{E} = \mathbf{T} \times \mathbf{R} = \mathbf{T}_x \mathbf{R}$ is the essential matrix if the cameras are calibrated and intrinsic parameters are known. $\mathbf{R}$ is the rotation matrix and $\mathbf{T}$ is the translation vector defined as previous, and $\mathbf{T}_x$ is a skew-symmetric matrix defined as
Figure 2.9: Epipolar geometry in an omnistereo system.

\[
T_x = \begin{bmatrix}
0 & -t_z & t_y \\
t_z & 0 & -t_x \\
-t_y & t_x & 0
\end{bmatrix}.
\]  

(2.24)

Epipolar Lines in Paracatadioptric Images

We can now see what the epipolar lines look like in paracatadioptric images. Assume \( \mathbf{n} = \mathbf{p}_1^T \mathbf{E} = [a_1, a_2, a_3]^T \). We have

\[ a_1 x_2 + a_2 y_2 + a_3 z_2 = 0. \]  

(2.25)

We substitute Equation (2.20) into Equation (2.25) and get

\[
\begin{bmatrix}
x'_2 \\
y'_2 \\
1
\end{bmatrix}
\begin{bmatrix}
a_3 & 0 & -\xi a_1 \\
0 & a_3 & -\xi a_2 \\
-\xi a_1 & -\xi a_2 & -\xi^2 a_3
\end{bmatrix}
\begin{bmatrix}
x'_2 \\
y'_2 \\
1
\end{bmatrix} = 0. \]  

(2.26)

This is an equation of a quadratic curve. More strictly, it is a circle centered at \((\xi a_1/a_3, \xi a_2/a_3)\) with a radius of \(\sqrt{\xi^2 + \frac{a_1^2 + a_2^2}{a_3^2}}\), or a straight line when \(a_3 = 0\). Hence, epipolar lines in paracatadioptric images essentially are epipolar circles or straight lines. The shape of an epipolar line depends on the angle between the epipolar plane and the mirror symmetry axes: it is a straight line when the mirror symmetry
axes are on the epipolar plane, otherwise is a circle. Examples of image points and their epipolar lines are shown in Figure 2.10 and Figure 2.11. Red points in the left image are the features of interest, with their correspondences on the green epipolar circles, which intersect at the two epipoles. The centers of these epipolar circles are on a line that is perpendicular to the line joining two epipoles [21].

![Figure 2.10: A paraomnidirectional image marked with feature points.](image1)

![Figure 2.11: A parao-image with correspondences and epipolar lines.](image2)

**Epipolar Lines in Hypercatadioptric Images**

Analogously substituting Equation (2.22) into Equation (2.25), we get the following equation for an epipolar line in a hypercatadioptric image.

\[
\begin{bmatrix}
  x_2' \\
  y_2'
\end{bmatrix}
\begin{bmatrix}
  (c^2 - 1)^2a_1^2 & (c^2 - 1)^2a_1a_2 & (c^2 - 1)(3c^2 - 1)a_1a_3f \\
  (c^2 - 1)^2a_1a_2 & (c^2 - 1)^2a_2^2 & (c^2 - 1)(3c^2 - 1)a_2a_3f \\
  (c^2 - 1)(3c^2 - 1)a_1a_3f & (c^2 - 1)(3c^2 - 1)a_2a_3f & 6c^2(c^2 - 1)a_3^2f^2
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  y_1
\end{bmatrix}
= 0.
\]

(2.27)

Hence, the epipolar lines in hypercatadioptric images might be ellipses, hyperbolas, parabolas, or lines. The shape of an epipolar line depends on the angle between the epipolar plane and the mirror symmetry axes [70].

**Estimate the Essential Matrix**

The essential matrix is determined by the translation and rotation between two cameras. Meanwhile, it can also be computed when a set of correspondences in a
pair of images is available. The estimated essential matrix is employed to recover the unknown relative positions of two cameras.

In the ideal case, meaning all correspondences are correctly matched and sufficiently accurate, only 6 points are needed to nonlinearly estimate the essential matrix, or 8 points for linear estimation. However, in practice, if the correspondences are obtained by a matching algorithm, it is hard to entirely prevent outliers. A robust method hence is needed in order to estimate an approximation of the essential matrix. The common way is using the linear least square method \cite{87} \cite{74}.

Expanding Equation (2.23) leads to the following linear system:

\[
\begin{bmatrix}
  x_1 x_2 & x_1 y_2 & x_1 z_2 & y_1 x_2 & y_1 y_2 & y_1 z_2 & z_1 x_2 & z_1 y_2 & z_1 z_2 \\
  e_{11} & e_{21} & e_{31} & e_{12} & e_{22} & e_{32} & e_{13} & e_{23} & e_{33}
\end{bmatrix} = 0, \quad (2.28)
\]

where, \(e_{ij}\) is the entry in \(i\)th row and \(j\)th column of \(E\). The system can be solved exactly when \(n\) pairs of correspondences \((n \geq 8)\) are given without noise. Due to noise and outliers, we use the least square method to find the best solution.

Let

\[
A = \begin{bmatrix}
  x_1 x_1 & x_1 y_1 & x_1 z_1 & y_1 x_1 & y_1 y_1 & y_1 z_1 & z_1 x_1 & z_1 y_1 & z_1 z_1 \\
  x_2 x_2 & x_2 y_2 & x_2 z_2 & y_2 x_2 & y_2 y_2 & y_2 z_2 & z_2 x_2 & z_2 y_2 & z_2 z_2 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  x_n x_n & x_n y_n & x_n z_n & y_n x_n & y_n y_n & y_n z_n & z_n x_n & z_n y_n & z_n z_n
\end{bmatrix}
\]

(2.29)

Then the least square problem is formalized as

\[
\text{Minimize } D = ||Ae||^2, \quad \text{subject to } ||e||^2 = 1,
\]

where \(e = [e_{11} e_{21} e_{31} e_{12} e_{22} e_{32} e_{13} e_{23} e_{33}]^T\).
Introducing a Lagrange multiplier $\lambda$, we attain the objective function as

$$O(e, \lambda) = ||Ae||^2 - \lambda(||e||^2 - 1) = e^T A^T A e - \lambda(e^T e - 1). \tag{2.30}$$

By differentiating with respect to $e$, it becomes an eigenvalue problem

$$A^T A e = \lambda e \tag{2.31}$$

where, $A^T A$ is the scatter matrix. The solution $e$ for this system is one of the eigenvector that corresponds to the minimum eigenvalue $\lambda_{\text{min}}$. $\lambda_{\text{min}}$ is the sum of the residuals.

### 2.3 Summary

In this chapter, we introduced some terminology and the classification of omnidirectional cameras, presented a projection model unifying both paracatadioptric and hypercatadioptric cameras, and derived an explicit formula for the stereo based 3D localization. This chapter is a short introduction to omnidirectional vision, upon which our work is built.
Chapter 3

Omnidirectional Camera Calibration

All geometries introduced in the previous chapter are described on the image plane. However, what we directly deal with are pixels in the quantized coordinates. There is a transformation between the image plane and the pixel coordinates, involving a number of parameters. The process of estimating the parameters of a camera system is called calibration, which is a necessary step in 3D reconstruction from 2D images. The quality of calibration can drastically affect the accuracy of reconstruction results.

In this chapter, motivated by the use of the omnistereo vision systems in our ASV mapping application, we focus on the calibration of paracatadioptric and hypercatadioptric cameras. I first introduce related work on calibration, and then present our calibration method. Following that, I numerically investigate the calibration performance using Monte Carlo simulation, which provides us a sense of the accuracy we can obtain when ground truth is unavailable. In particular simulation allows analysis of range estimation performance with respect to individual and combinations of parameters. Finally, the calibration results of our cameras are presented and validated by a 3D reconstruction application.
3.1 Related Work

The problem of calibrating perspective cameras has been well studied [88], while the study of calibration for different kinds of omnidirectional cameras is still ongoing. Most omnidirectional camera calibration techniques performed better on paracatadioptric cameras than on other types because the paracatadioptric projection model is relative simple.

In reviewing the literature on both perspective and omnidirectional cameras, all calibration techniques can be roughly classified into three categories [42], according to their use of scene knowledge.

1. **Calibration with exact scene knowledge.** This type of methods is chosen when one has access to the camera and can place a known object in the scene. The object used for calibration is called the *calibration rig* that may be a planar checkerboard [88], a checkerboard-textured cube [42], or a checkerboard-textured barrel [72]. The coordinate values of a certain number of points in this object are known with high accuracy. Most calibration toolboxes available online belong to this type of techniques.

2. **Calibration with partial scene knowledge.** This type of methods is performed in the presence of partial scene knowledge. Namely, there are some objects whose geometries in 3D space are partially known. For instance, man-made objects contain planar surfaces and parallel lines. In [19], Geyer *et al.* use parallel lines as constraints to estimate intrinsic parameters for a paracatadioptric camera.

3. **Calibration without scene knowledge.** Techniques in this category do not use any calibration object. They estimate parameters based on correspondences [41] [34] in pairs of images which are taken by the camera at different positions. Although this type of techniques is more convenient since no prior knowledge is needed, it has constraints on the camera displacements and its precision is generally worse than those obtained using photogrammetric techniques.
3.2 Calibration Method

The problem of calibration is generally formalized as an optimization problem that estimates the parameters appearing in a camera projection model by minimizing a cost function. Most calibration techniques differentiate themselves mainly by the chosen camera projection model, the cost function to be optimized, the initial parameters, and the optimization procedure.

3.2.1 Projection Model

Taking into account misalignment and distortion, there are five coordinate systems, four transformations and two projections in total, as indicated in Figure 3.1 [41]. The sequential transformations and projections are:

Figure 3.1: The general projection model for an omnidirectional camera.

1. **The transformation from the world coordinate system to the mirror coordinate system, denoted as** $\mathcal{F}_1$. A 3D point $P_w = [X_w, Y_w, Z_w]^T$ in the world coordinates has to be transformed into the point $P_m = [X_m, Y_m, Z_m]^T$ in the mirror coordinates following

$$P_m = RP_w + T,$$  

(3.1)
where \( \mathbf{R} \) and \( \mathbf{T} \), the rotation matrix and the translation vector between the two coordinate systems, are called the *extrinsic parameters*.

2. The projection from the point \( \mathbf{P}_m \) in the mirror coordinates to the point \( \mathbf{P}_{m1} \) on the mirror surface, denoted as \( \mathcal{P}_1 \).

\[
\mathbf{P}_{m1} = \frac{\zeta}{\epsilon Z_m + \sqrt{X_m^2 + Y_m^2 + Z_m^2}} \mathbf{P}_m
\]

(3.2)

The parameters \( \zeta \) and \( \epsilon \) depend on the mirror’s profile. \( \zeta = 2h \) and \( \epsilon = 1 \) for a parabolic mirror. \( \zeta = \frac{b^2}{a} \) and \( \epsilon = \frac{c}{a} \) for a hyperbolic mirror.

3. The transformation from the point \( \mathbf{P}_{m1} \) in the mirror coordinates to the point \( \mathbf{P}_1 = [X, Y, Z]^T \) in the camera coordinates, denoted as \( \mathcal{F}_2 \), which is

\[
\mathbf{P}_1 = \mathbf{R}_{mc} \mathbf{P}_{m1} + \mathbf{T}_{mc},
\]

(3.3)

where \( \mathbf{R}_{mc} \) and \( \mathbf{T}_{mc} \) are the rotation matrix and the translation vector between the two coordinate systems.

In an ideal camera model, these two coordinates are aligned in order to attain a single viewpoint. However, due to the manufacturing inaccuracies, there might be small alignment errors to be considered.

4. The projection for the point \( \mathbf{P}_1 \) in the camera coordinates to the point \( \mathbf{p} = [x, y]^T \) in the image plane, denoted as \( \mathcal{P}_2 \).

\[
\begin{bmatrix}
  x \\
  y
\end{bmatrix} = \lambda \begin{bmatrix}
  X \\
  Y
\end{bmatrix}
\]

(3.4)

The parameter \( \lambda \) depends on the type of cameras,

\[
\lambda = \begin{cases} 
  1 & \text{for paracatadioptric camera} \\
  -\frac{f}{2z} & \text{for hypercatadioptric camera}
\end{cases}
\]

(3.5)

5. The transformation from the image plane to the pixel coordinate system, denoted as \( \mathcal{F}_3 \). When the point is projected onto the image plane its position is represented in the pixel coordinates as \( \mathbf{p}' = [u, v]^T \). Assume the origin of the image plane is located at \( \mathbf{c} = [u_0, v_0]^T \) in the pixel coordinate system.
system, the scale factors along the $u$ and $v$ axes are $k_u$ and $k_v$, and the skew between the two pixel axes is $s_0$, then the pixel coordinates in the image may be computed by

$$p' = Kp + c \quad \text{with} \quad K = \begin{bmatrix} k_u & s_0 \\ 0 & k_v \end{bmatrix}. \quad (3.6)$$

6. The transformation from the perfect image to the distorted one, denoted as $F_4$. The main source of distortion is the imperfection of the lens shape, modeled by radial distortion [80]. Let $p_d = [u_d, v_d]^T$ be the distorted pixel point, then

$$p_d = p' + (p' - c)(k_1\rho^2 + k_2\rho^4), \quad (3.7)$$

where $k_1$ and $k_2$ are the coefficients of the radial distortion, and $\rho = \sqrt{x^2 + y^2}$.

Note that in the projection $P_2$, the focal length and minus sign appearing in Equation (3.5) is transferred to the transformation $F_3$ when we composite $P_2$ and $F_3$ together. Thus, the values depend on the products $fk_u$, $fk_v$, and it is not possible to discriminate between a change of the focal length and a change of the scale factors on the pixel axes. For this reason, we introduce the new parameters $\alpha_u = -fk_u$, $\alpha_v = -fk_v$, and $s = -fs_0$, and get

$$A = \begin{bmatrix} \alpha_u & s \\ 0 & \alpha_v \end{bmatrix}. \quad (3.8)$$

The parameters in $A$ and $c$ are called the intrinsic parameters.

The calibration of an omnidirectional camera requires estimation of all or part of the following parameters:

- Extrinsic parameters: $R$ and $T$
- Intrinsic parameters: $\alpha_u$, $\alpha_v$, $s$, $u_0$, $v_0$.
- Mirror parameters: $\varsigma$ and $\epsilon$.
- Misalignment parameters: $R_{mc}$ and $T_{mc}$.
- Distortion coefficients: $k_1$ and $k_2$.  

27
There are other projection models available for omnidirectional cameras. For example, in [72] and [47], researchers build unified models fitting both hyperbolic and parabolic cameras. Their models in essence are the same as ours, with the same number of parameters. In [61], Davide et al. introduced a polynomial form to approximate the projection model for all omnidirectional cameras. That model fits the parabolic camera exactly using a two-degree polynomial, while a polynomial of degree five is required to approximate the hyperbolic camera, hence the calibration of hypercatadioptric cameras are not accurate enough.

### 3.2.2 A Calibration Rig

As in other calibration methods, we use a checkerboard as the calibration rig. The first use of a planar checkerboard is by Zhang [88] for a perspective camera calibration. Figure 3.2 shows an example of the images used in calibration.

![Figure 3.2: An image used in calibration.](image.png)

Since we are free to choose the world coordinate system, we fix it on the checkerboard plane so that all points on the board have $Z_w = 0$, and align the axes $X_w$ and $Y_w$ along with two perpendicular grid lines respectively. With the known size of each grid, it is possible to know the world coordinate values for each corner point.
3.2.3 The Cost Function

As introduced in Section 3.2.1, given a point in the world coordinates, the pixel coordinates are obtained by

\[ p_d = F_4 \circ F_3 \circ P_2 \circ F_2 \circ P_1 \circ F_1 (P_w) \]  

(3.9)

In the calibration, we aim to estimate the parameters in this nonlinear function such that the sum of the difference between the calculated pixel value and the observed pixel value for corner points is minimized. Hence the cost function is designed as follows.

**Case 1: Calibrate a Single Paracatadioptric or Hypercatadioptric Camera**

Let \( X = \{ \alpha_u, \alpha_v, s, u_0, v_0, \epsilon, R_{mc}, T_{mc}, k_1, k_2 \} \). Given \( m \) image frames, each of which has \( n \) corner points, the calibration is formalized as

\[
\text{arg min}_{X \cup \{ R_i, T_i \}} \sum_{i=1}^{m} \sum_{j=1}^{n} \| p_{dij} - \hat{p}_{ij} \|^2
\]  

(3.10)

where, \( p_{dij} = [u_{dij}, v_{dij}]^T \) is the calculated pixel value of the \( j \)th point in the \( i \)th image by Equation (3.9), and \( \hat{p}_{ij} = [\hat{u}_{ij}, \hat{v}_{ij}]^T \) is the observation value measured from the image.

**Case 2: Calibrate Two Hyperbolic Cameras**

Let \( X_t = \{ \alpha_{ut}, \alpha_{vt}, s_t, u_{0t}, v_{0t}, \epsilon_t, R_{mc_t}, T_{mc_t}, k_{1t}, k_{2t} \} \) be the parameters for the camera \( t, t = 1, 2 \). Then the calibration is formalized as

\[
\text{arg min}_{X_1 \cup X_2 \cup \{ R_{12}, T_{12} \} \cup \{ R_{i}, T_{i} \}} \sum_{i=1}^{m} \sum_{j=1}^{n} \| p_{dij} - \hat{p}_{ij} \|^2
\]  

(3.11)

where \( R_{12} \) and \( T_{12} \) are the rotation matrix and the translation vector between the two cameras.

3.2.4 Initialization and Calibration Steps

Since the projection model is a nonlinear function, the calibration is best done using an iterative numerical optimization method such as the Levenberg-Marquardt
(LM) method [45] [38]. The LM method is a standard technique for nonlinear least-square problems and can be thought of a combination of the steepest descent and the Gauss-Newton method. In a nonlinear optimization problem, it is critical to initialize the parameters properly when using an iterative method, because it is highly likely for the minimization procedure to converge to a local minimum when the number of parameters is large. In our case, we have $15 + 6 \times m$ (m is the number of images) parameters to estimate.

The calibration procedure is:

1. Print a grid pattern and attach it to a planar board.
2. Take several images of the checkerboard by the hypercatadioptric camera.
3. Detect the corner points in the images.
4. Manually point the center of an image to get initial $u_0$ and $v_0$.
5. Calibrate the projection model with $\alpha_u = \alpha_v$, $s = 0$, ignoring the misalignment and the distortion errors. From experiments, we know that the errors in the initial values for $\alpha_u$ and $\alpha_v$ do not affect the convergence results in this simplified camera model.
6. Calibrate the projection model that ignores the misalignment and the distortion errors by taking previous results as initial values for corresponding parameters.
7. Calibrate the entire projection model by taking previous results as initial values for corresponding parameters.
3.3 Simulated Experiments

Although that using rigs to calibrate is more accurate than other methods, there are several factors affect the calibration accuracy [37] [88] [68]:

- The accuracy of the 3D object points. We usually assume the 3D points are free of error, but practically the patterns may be distorted slightly during manufacture.

- The error of measuring 2D image points. How well we can measure 2D points in the image depends on the precision of the corner detection algorithm or the accuracy of the manual corner selection.

- The imaging geometry. Choice of rig placement and orientation can affect calibration performance.

- The number of image frames used in calibration.

- The calibration model, including the projection model and the cost function used to estimate the parameters.

Below we look at the influence of the above factors on the accuracy of the intrinsic calibration parameters. In order to compare to ground truth of the parameters and isolate each effect, we conduct experiments on simulated data. Meanwhile, the sensitivity to noise is examined using Monte Carlo simulation, by adding Gaussian noise with 0 mean and $\sigma$ standard deviation to some of the input data. The reason for using Monte Carlo simulation is that it is difficult to investigate the sensitivity analytically because of the non-linearity of the omnidirectional imaging model.

3.3.1 Synthetic Data

The properties of the simulated hypercatadioptric camera and the simulated virtual planes are listed in Table 3.2. The planes are also plotted in Figure 3.3: the first four are these in blue, and the second four are in red and the remaining are in green. The resolution of images is $1280 \times 1024$ pixels. Each plane contains $6 \times 8$ cells with the size of $30mm \times 30mm$. 
<table>
<thead>
<tr>
<th>Camera Properties</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Focal Length:</td>
<td></td>
</tr>
<tr>
<td>Principle Point:</td>
<td></td>
</tr>
<tr>
<td>Mirror Parameters:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image Planes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(r_1 = [0^\circ, 0^\circ, 90^\circ]^T) (t_1 = [-500, -100, 300]^T mm)</td>
</tr>
<tr>
<td>2</td>
<td>(r_2 = [30^\circ, 0^\circ, 100^\circ]^T) (t_2 = [-100, -500, 300]^T mm)</td>
</tr>
<tr>
<td>3</td>
<td>(r_3 = [0^\circ, 0^\circ, 70^\circ]^T) (t_3 = [500, -100, 300]^T mm)</td>
</tr>
<tr>
<td>4</td>
<td>(r_4 = [0^\circ, 15^\circ, 90^\circ]^T) (t_4 = [100, 500, 300]^T mm)</td>
</tr>
<tr>
<td>5</td>
<td>(r_5 = [0^\circ, 30^\circ, 60^\circ]^T) (t_5 = [-400, -200, 400]^T mm)</td>
</tr>
<tr>
<td>6</td>
<td>(r_6 = [60^\circ, 0^\circ, 100^\circ]^T) (t_6 = [-200, -400, 400]^T mm)</td>
</tr>
<tr>
<td>7</td>
<td>(r_7 = [60^\circ, 30^\circ, 70^\circ]^T) (t_7 = [400, -200, 400]^T mm)</td>
</tr>
<tr>
<td>8</td>
<td>(r_8 = [0^\circ, 60^\circ, 100^\circ]^T) (t_8 = [200, 400, 400]^T mm)</td>
</tr>
<tr>
<td>9</td>
<td>(r_9 = [45^\circ, 45^\circ, 0^\circ]^T) (t_9 = [300, 300, 200]^T mm)</td>
</tr>
<tr>
<td>10</td>
<td>(r_{10} = [0^\circ, 45^\circ, 0^\circ]^T) (t_{10} = [-300, 300, 200]^T mm)</td>
</tr>
<tr>
<td>11</td>
<td>(r_{11} = [45^\circ, 0^\circ, 0^\circ]^T) (t_{11} = [300, -300, 200]^T mm)</td>
</tr>
<tr>
<td>12</td>
<td>(r_{12} = [45^\circ, 45^\circ, 90^\circ]^T) (t_{12} = [-300, -300, 200]^T mm)</td>
</tr>
</tbody>
</table>

Table 3.1: Simulated Data

Figure 3.3: The simulated planes in calibration.
3.3.2 Evaluation Criteria

The criteria to measure the impact of the above factors on calibration are

1. The back projection error.
2. The relative errors of $\alpha$ and $\beta$, i.e. $|\Delta\alpha|/|\alpha|$ and $|\Delta\beta|/|\beta|$.
3. The absolute errors of $u_0$ and $v_0$, i.e. $|\Delta u_0|$ and $|\Delta v_0|$.

3.3.3 Calibration Sensitivities

Sensitivities w.r.t Perturbation of 3D Points

In this experiment, we look at the calibration results with respect to the measurement errors introduced in the 3D calibration points using the first four simulated planes. Gaussian noise is added to the position of each 3D point. The noise level is varied from 0.05 pixels to 0.5 pixels. For each noise level, we perform 100 independent calibration trials and compute the average and the standard deviation value for each parameter. The results are shown in Figure 3.4, 3.5 and 3.6.

Sensitivities w.r.t Perturbation of Corner Detection

With the same setup, we add different levels of Gaussian noise to the corners detected in the image planes to study the effect of noise on calibration accuracy. The results are shown in Figure 3.7, 3.8 and 3.9.

Performance w.r.t the Number of Planes

In this case we set a noise level of $\sigma = 0.3$ to image points and vary the number of image frames from 2 to 11. The results are shown in Figure 3.10, 3.11 and 3.12.

Performance w.r.t the Depth Range of planes

In this experiment, we aim to explore the impact of the depth range and orientation of planes on the calibration. Two groups of image planes are tested: the first group includes Plane 1 ~ 4, and the second includes Plane 5 ~ 8. The results are shown in Figure 3.13, 3.14 and 3.15.
Figure 3.4: Back-projection error and the standard deviation vs. the noise level in the checkerboard.

Figure 3.5: Relative error of $\alpha, \beta$, the mean values and standard deviations vs. the noise level in the checkerboard.

Figure 3.6: Absolute error of $u_0, v_0$, the mean values and standard deviations vs. the noise level in the checkerboard.
Figure 3.7: Back-projection error and the standard deviation vs. the noise level in corner detection.

Figure 3.8: Relative error of $\alpha, \beta$, the mean values and standard deviations vs. the noise level in corner detection.

Figure 3.9: Absolute error of $u_0, v_0$, the mean values and standard deviations vs. the noise level in corner detection.
Figure 3.10: Back-projection error and the standard deviation vs. the number of planes.

Figure 3.11: Relative error of $\alpha, \beta$, the mean values and standard deviations vs. the number of planes.

Figure 3.12: Absolute error of $u_0, v_0$, the mean values and standard deviations vs. the number of planes.
Figure 3.13: Back-projection error and the standard deviation using two groups of image planes.

Figure 3.14: Relative error of $\alpha, \beta$, the mean values and standard deviations using two groups of image planes.

Figure 3.15: Absolute error of $u_0, v_0$, the mean values and standard deviations using two groups of image planes.
3.3.4 Discussion

The simulations demonstrate that

• Generally speaking, calibration errors increase when the noise levels increase.

• An increase of the number of image planes does not improve the calibration performance when the noise level is fixed and there are enough points on each plane.

• When the checkerboard spans a larger range of depth values, the focal length \( \alpha, \beta \) is recovered more accurately.

• The backprojection error is less than 1 pixel in most of cases.

• The relative errors of \( \alpha, \beta \) are smaller than 10% in most of cases when the noise level is lower than an half pixel.

• The estimation of the image center \((u_0, v_0)\) is relatively unstable. The error is from an half pixel to more than ten pixels.

The latter two observations provide us the order of calibration errors which are applied to analyze range estimation performance in Chapter 4.

3.4 Real Experiments

3.4.1 Calibration Results

Table 3.2 lists the parameters of one paracatadioptric camera and two hypercatadioptric cameras used in the omnivision system.

3.4.2 Validation

In order to validate the calibrated parameters, we test them in an indoor localization experiment. In this experiment, two hypercatadioptric cameras are placed on a stable frame. The frame is moved, hence images are taken by two cameras at
<table>
<thead>
<tr>
<th>Calibration Results of Paracatadioptric Camera</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focal Length:</td>
</tr>
<tr>
<td>( \alpha = 16.07472 ) ( \beta = 16.06252 ),</td>
</tr>
<tr>
<td>Principle Point:</td>
</tr>
<tr>
<td>( u_0 = 633.21 ) ( v_0 = 588.07 )</td>
</tr>
<tr>
<td>Mirror Parameters:</td>
</tr>
<tr>
<td>( h = 16.7 )</td>
</tr>
<tr>
<td>Calibration Results of Hypercatadioptric Camera 478</td>
</tr>
<tr>
<td>Focal Length:</td>
</tr>
<tr>
<td>( \alpha = 1424.0 ) ( \beta = 1422.6 ),</td>
</tr>
<tr>
<td>Principle Point:</td>
</tr>
<tr>
<td>( u_0 = 625.79 ) ( v_0 = 503.72 )</td>
</tr>
<tr>
<td>Mirror Parameters:</td>
</tr>
<tr>
<td>( a = 29.01 ) ( b = 24.13 )</td>
</tr>
<tr>
<td>Calibration Results of Hypercatadioptric Camera 480</td>
</tr>
<tr>
<td>Focal Length:</td>
</tr>
<tr>
<td>( \alpha = 1358.9 ) ( \beta = 1360.8 ),</td>
</tr>
<tr>
<td>Principle Point:</td>
</tr>
<tr>
<td>( u_0 = 647.51 ) ( v_0 = 506.96 )</td>
</tr>
<tr>
<td>Mirror Parameters:</td>
</tr>
<tr>
<td>( a = 28.09 ) ( b = 23.41 )</td>
</tr>
</tbody>
</table>

Table 3.2: Calibration Results

different positions. The positions of landmarks are recovered from each single camera and also from the stereo cameras respectively. As shown in Figure 3.18, green crosses are manually measured positions, red points are the localization results from one hypercatadioptric camera, blue points are from the other hypercatadioptric camera, and yellow points are from the two cameras. We can see that the results obtained by each single camera are much more accurate than those from the stereo cameras. Since measurement errors on stereo configuration and movement of the frame are in the same order, the main differences are from calibration errors.

This experiment validates the intrinsic parameters estimated by calibration algorithm, but also implies that range estimation using two omnidirectional cameras are more sensitive to discalibration than that of using a single camera. The reason will be further illustrated in Chapter 4.
Figure 3.16: An image taken by the first hypercatadioptric camera.

Figure 3.17: An image taken by the second hypercatadioptric camera.

Figure 3.18: 3D reconstruction results.
3.5 Summary

In this chapter, we implemented a calibration algorithm for paracatadioptric and hypercatadioptric cameras. Monte Carlo simulation was tested in order to evaluate the algorithm and get a sense of calibration accuracy. The cameras used in our application were calibrated. An indoor localization experiment is conducted to validate the calibrated parameters. This experiment also shows that range estimation using a single moving camera is more stable to calibration errors than that of using two cameras.
Chapter 4

Performance Analysis of Omnistereo Vision

Performance is a primary concern in any omnistereo vision system. Performance analysis in this chapter includes an investigation of the relationship between the maximum detectable range and the stereo configuration, together with an exploration of the sensitivity of 3D localization with respect to inaccuracies introduced during correspondence matching, stereo configuration, and camera calibration.

Performance analysis offers insights into the capabilities and limitations of our vision system. When designing and assembling an omnistereo system on an ASV, we should consider the following factors: the payload limitation of the robot, occlusion of the cameras’ fields of view, the maximum detectable range, and the attainable accuracy of range estimation. A trade-off exists when choosing a stereo configuration, in that accurate range estimation requires a large baseline while a small baseline is preferable for accurate correspondence matching. Performance analysis guides the choice of stereo configurations, which impacts the robot’s navigation in outdoor mapping tasks.

In this chapter I first give a short review of the related work on performance analysis. Then, I estimate the maximum detectable range, analyze the accuracy of 3D localization when the stereo vision system suffers from several kinds of noise. I further illustrate the sensitivity distributions for some special cases which simulate real application scenarios of our omnivision system.
4.1 Related Work

A survey of the literature on omnidirectional vision shows that previous research has primarily focused on the design of sensors [50] [85], the calibration of omnidirectional cameras [60], and the problems related to structure or motion from omnistereo, including omnistereo geometries [70], correspondence matching algorithms [20], and range estimation [24]. There is relatively little work investigating the performance of such omnistereo systems.

In the work done by Ishiguro et al. [32], they use a pinhole camera swiveling about a vertical axis to obtain omnidirectional images. The direction-dependent uncertainty in range estimation is analyzed with respect to the uncertainty in bearing angle. The results show that the uncertainty is small in the direction perpendicular to the baseline while large in the baseline direction.

Ishiguro et al. [66] also estimated the precision of an N-ocular omnistereo system used for human tracking. The human as a target is localized in the overlapped region by considering both observation errors on bearing angles and an error of the human model. Ishiguro’s work [32] [66] presented the uncertainties more graphically than algebraically.

Shum et al. [64] designed an omnivergent stereo system that is a virtual sensor for optimal 3D reconstruction. They studied the performance of the case where an omnidirectional camera moves within a circular region on a plane, and concluded that the reconstruction accuracy is optimized for two cameras with a maximum vergence angle.

In contrast to the above-reviewed work, Zhu et al. [89] has done more elaborate work on the analysis of the accuracy in range estimation. The authors numerically analyzed the error characteristics of an adjustable omnistereo system with respect to mutual calibration error, stereo matching error, and triangulation error. The result of performance analysis is used to derive the rules of optimal view planning.

All of these studies take an omnidirectional camera as a bearing sensor. Both range estimation and performance analysis are based on triangulation in terms of bearing angles. In addition, they assume that cameras are assembled or moving on a flat plane and no relative rotation exists.
4.2 Maximum Detectable Range

The maximum detectable range is a basic property of a stereo vision system. It is valuable to know the range limit on the visual field, especially when the system is applied in an open area. In contrast to a conventional pinhole stereo system, whose maximum detectable range is determined by $bf/\delta$ [59] when the baseline $b$, the focal length $f$ and the pixel size $\delta$ are given, the maximum detectable range of an omnistereo system is not explicit. In fact, it varies in different directions. The space bounded by the maximum range is named the detectable region in this work. We here explore the relationship between the detectable region and the stereo configuration.

4.2.1 Derivation of the Maximum Detectable Range

The detectable region is determined by the stereo configuration and the cameras’ intrinsic parameters. The image plane of a CCD camera is quantized into integral pixel coordinates. Each scene point projected into the image plane is rounded to the nearest pixel. Therefore, in order to locate a scene point, the disparity of the projections in a pair of images must be greater than the size of a pixel. (Here, as other papers did, we consider the accuracy on pixel level instead of subpixel level to get a rough estimation.) The disparity is commonly defined as

$$d = p_1' - p_2'$$  \hspace{1cm} (4.1)

where, $p_1' = [x_1', y_1']^T$ and $p_2' = [x_2', y_2']^T$ are pairwise correspondences on the image planes as defined previously. Regarding the constraint of the minimal disparity, we say that a 3D point is undetectable if its disparity satisfies the following inequality

$$|d|_\infty < s_p$$  \hspace{1cm} (4.2)

where $|d|_\infty$ is the $L^\infty$ norm of $d$, and $s_p$ is the pixel size. For ease of analysis, we assume that the width and the height of a pixel are equal to each other.

Note that the disparity caused by a pure rotation about $Z$ axis does not contribute to the detection capability. The reason is simply illustrated by a special case in which a single moving camera is purely rotated at two time instants. In this
case, any 3D point is unobservable no matter how large its disparity is. Therefore, the rotation is compensated for when we estimate the detectable region. The disparity with rotation compensation is defined as

\[ d = p'_1 - R_{2 \times 2}p'_2, \] (4.3)

where, \( R_{2 \times 2} \) is the submatrix of \( R \). To simplify the analysis, we assume that there is only a rotation about the \( Z \) axis. Hence,

\[ R_{2 \times 2} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \] (4.4)

Following the projection geometry presented in Section 2.2, we get the following theorem.

**Theorem 2.** Given an omnistereo configuration with a translation \( T = [t_x, t_y, t_z]^T \) and a rotation \( \theta \) about the axis \( Z_1 \), a point \( P = [X, Y, Z]^T \) is undetectable if it satisfies

\[ |d|_\infty = \max\{|C_1 X - C_2 (X - t_x)|, |C_1 Y - C_2 (Y - t_y)|\} < s_p, \] (4.5)

where

\[ C_1 = \frac{f \xi}{\zeta Z + |P|}, \] (4.6)

\[ C_2 = \frac{f \xi}{\zeta (Z - t_z) + |P - T|}. \] (4.7)

**Proof.** According to the projection model in Equation (2.10), we have

\[ \begin{bmatrix} x'_1 \\ y'_1 \end{bmatrix} = \frac{f \xi}{\zeta Z + |P|} \begin{bmatrix} X \\ Y \end{bmatrix}. \] (4.8)

To calculate the projection point \( p'_2 = [x'_2, y'_2]^T \) in the second image, we first transform \( P \)'s coordinates from the first camera coordinate system into the second one by

\[ P^{2nd} = R^{-1}(P - T), \] (4.9)

where \( P^{2nd} = [X^{2nd}, Y^{2nd}, Z^{2nd}]^T \) is \( P \)'s position in reference to the second camera. Then, we get

\[ \begin{bmatrix} x'_2 \\ y'_2 \end{bmatrix} = \frac{f \xi}{\zeta Z^{2nd} + |P^{2nd}|} \begin{bmatrix} X^{2nd} \\ Y^{2nd} \end{bmatrix}. \] (4.10)
Combining Equation (4.8), (4.10), and (4.3), we obtain

\[ d = C_1 \begin{bmatrix} X \\ Y \end{bmatrix} - C_2 \begin{bmatrix} X - b_x \\ Y - b_y \end{bmatrix}. \]  \hspace{1cm} (4.11)

Hence,

\[ |d|_{\infty} = \max \{|C_1X - C_2(X - t_x)|, |C_1Y - C_2(Y - t_y)|\}. \]  \hspace{1cm} (4.12)

4.2.2 Disparity Distributions

In order to demonstrate the detectable region, we plot the disparity distributions of two special cases related to our real application, in which the ASV navigates in a lake or a bay trying to map the shoreline using omnistereo vision.

- **A paracatadioptric camera moving on a horizontal plane.** This case uses a single paracatadioptric omnidirectional camera, so the stereo vision is obtained when the ASV moves on the water surface. The parameter of the camera is \( \xi = 2h = 0.0334m \). We assume for illustration that the translation is \( T = [0.6, 0.8, 0]^T m \) and yaw is \( \theta = 30^\circ \).

- **Two hypercatadioptric cameras assembled along a vertical axis.** The second case uses two fixed hypercatadioptric cameras which are set up with the baseline \( T = [0, 0, -0.711]^T m \) and without rotation. We assume for illustration that these two cameras have the same intrinsic parameters: \( \xi = 0.2667m, \zeta = 1.0350m, \) and \( f = 5mm \).

For both cases, the pixel size is \( s_p \approx 6.4 \times 10^{-5}m \). Referring to the shoreline positions in real scenarios, we confine the points of interest on the horizontal plane \( Z = 1m \). The distributions of disparity for the two cases are shown in Figure 4.1 and 4.2 respectively.

In view of the disparity values, we see that the detectable region is the part inside the yellow contours. For the first case, the detectable region is symmetric about the translation and it is about 400 meters in the region perpendicular to the
ASV’s trajectory, while only tens of meters in-line with the velocity of the ASV. For the second case, the detectable range is about 100 \(\sim\) 150 meters, shorter than the first case due to the smaller baseline. The detectable region is symmetric about the origin and \(X, Y\) axes.

![Image](image1.png)

**Figure 4.1:** A disparity distribution for case 1.

![Image](image2.png)

**Figure 4.2:** A disparity distribution for case 2.
4.3 Localization Sensitivity Analysis

In addition to the detectability, the 3D localization accuracy in practical application is another performance metric we are interested in. In the following we investigate the impact on 3D localization resulting from noise in feature tracking, stereo configuration, and camera calibration.

4.3.1 Derivation of $\Delta P$

Let the reconstructed 3D point $P$ be a function of a parameter set $\{t_1, t_2, ..., t_n\}$. That is,

$$P = F(t_1, t_2, ..., t_n). \quad (4.13)$$

Then, the sensitivity of $P$ with respect to a disturbance in the parameter $t_i$ can be computed by partial differentiation of Equation (4.13) as

$$\Delta P_{t_i} = \frac{\partial F}{\partial t_i} \Delta t_i \quad (4.14)$$

or discretely by

$$\Delta P_{t_i} = F(t_1, t_2, ..., t_i + \Delta t_i, ..., t_n) - F(t_1, t_2, ..., t_i, ..., t_n). \quad (4.15)$$

When using Formula (4.14), we expect to obtain a concise equation from which we can determine the relationship between the sensitivity and the parameters easily. However, $\Delta P$ calculated as equation(4.14) is highly complicated, since it depends not only on the correspondence matching, stereo configuration, but also on the calibration results. To avoid tedious equations, we adopt Formula (4.14) only to derive the localization sensitivity w.r.t matching and configuration errors for paracatadioptric cameras that are relatively concise. For other cases, such as when using hypercatadioptric cameras and considering calibration errors, we use Formula (4.15) to calculate and plot error distributions. In both cases, I transform $\Delta P$ from a function of pixels to a function of 3D positions in order to observe the relationship between the sensitivity and the 3D locations of scene points.

I first compute the 3D localization error $\Delta P$ w.r.t an error $\Delta t$ existing for an arbitrary parameter $t$ in Equation (2.17) as follows.

$$\Delta P_t = F \frac{\partial p_1}{\partial t} \Delta t + p_1 \frac{\partial F}{\partial t} \Delta t. \quad (4.16)$$
Let
\[ A_1 = (c \times b) \cdot (a \times b) \] (4.17)
\[ A_2 = |a \times b|^2 = (a \times b) \cdot (a \times b). \] (4.18)

Then,
\[ F = \frac{(c \times b) \cdot (a \times b)}{|a \times b|^2} = \frac{A_1}{A_2}, \] (4.19)

and,
\[
\frac{\partial F}{\partial t} = \frac{\partial F}{\partial A_1} \frac{\partial A_1}{\partial t} + \frac{\partial F}{\partial A_2} \frac{\partial A_2}{\partial t} = \frac{1}{A_2} \frac{\partial}{\partial t} \left[ (c \times b) \cdot (a \times b) \right] - \frac{A_1}{A_2^2} \frac{\partial}{\partial t} \left[ (a \times b) \cdot (a \times b) \right]
\]
\[
= \frac{1}{A_2^2} \left[ \frac{\partial}{\partial t} \left( (c \times b) - (a \times b) + \frac{\partial}{\partial t} (a \times b) \right) \cdot (a \times b) + (c \times b) - \frac{\partial}{\partial t} (a \times b) \cdot (a \times b) \right]
\]
\[
= \frac{2A_1}{A_2^2} \left[ \frac{\partial}{\partial t} (a \times b) \cdot (a \times b) \right].
\] (4.20)

This derivation is used in the analysis of the localization sensitivity for paracata-
dioptric cameras.

4.3.2 Sensitivity w.r.t. Matching Error

Given a point \( p'_1 \) in the first image, its correspondence \( p'_2 \) in the second image is obtained by a matching algorithm. In this sense, there is no error for \( p'_1 \). However, due to the changes in view direction and illumination for two images, matching algorithms cannot always locate the correspondence exactly. The displacement between the real correspondence and the one obtained by matching methods is the matching error [89], denoted as \((\Delta x'_2, \Delta y'_2)\). The 3D localization error resulting from the matching error is denoted as \( \Delta P_M \).
We investigate $\Delta P_M$ for two omnistereo vision systems. The first one is using a single paracatadioptric camera that moves on a horizontal plane. The second simulates two fixed hypercatadioptric cameras assembled along a vertical line.

**Case 1: A Paracatadioptric Camera Moving on A Horizontal Plane**

We calculate $\Delta P_M$ using partial differentiation as

$$\Delta P_M = p_1 \left( \frac{\partial F}{\partial x'_2} \Delta x'_2 + \frac{\partial F}{\partial y'_2} \Delta y'_2 \right). \tag{4.21}$$

Substituting $t$ in Equation (4.20) by $x'_2$, we get

$$\frac{\partial F}{\partial x'_2} \Delta x'_2 = \frac{1}{A_2} \left[ \left( \frac{\partial c}{\partial x'_2} \times b + c \times \frac{\partial b}{\partial x'_2} \right) \cdot (a \times b) + (c \times b) \cdot \left( \frac{\partial a}{\partial x'_2} \times b + a \times \frac{\partial b}{\partial x'_2} \right) \right]$$

$$- \frac{2A_1}{A_2} \left[ (a \times \frac{\partial b}{\partial x'_2}) \cdot (a \times b) \right] \Delta x'_2$$

$$= \frac{1}{A_2} \left[ (c \times \frac{\partial b}{\partial x'_2}) \cdot (a \times b) + (c \times b) \cdot (a \times \frac{\partial b}{\partial x'_2}) \right] - \frac{2A_1}{A_2} \left( (a \times \frac{\partial b}{\partial x'_2}) \cdot (a \times b) \right) \Delta x'_2. \tag{4.22}$$

Analogously,

$$\frac{\partial F}{\partial y'_2} \Delta y'_2 = \frac{1}{A_2} \left[ (c \times \frac{\partial b}{\partial y'_2}) \cdot (a \times b) + (c \times b) \cdot (a \times \frac{\partial b}{\partial y'_2}) \right] - \frac{2A_1}{A_2} \left( (a \times \frac{\partial b}{\partial y'_2}) \cdot (a \times b) \right) \Delta y'_2. \tag{4.23}$$

Since

$$\frac{\partial b}{\partial x'_2} = R \begin{bmatrix} 1 \\ 0 \\ \frac{x'_2}{\xi} \end{bmatrix}$$

and

$$\frac{\partial b}{\partial y'_2} = R \begin{bmatrix} 0 \\ 1 \\ -\frac{y'_2}{\xi} \end{bmatrix}, \tag{4.24}$$

we get,

$$\Delta P_M = p_1 \left( \frac{\partial F}{\partial x'_2} \Delta x'_2 + \frac{\partial F}{\partial y'_2} \Delta y'_2 \right)$$

$$= p_1 \left( \frac{1}{A_2} \left[ (c \times R \Delta p_2) \cdot (a \times b) + (c \times b) \cdot (a \times R \Delta p_2) \right] - \frac{2A_1}{A_2} \left( (a \times R \Delta p_2) \cdot (a \times b) \right) \right), \tag{4.25}$$

where $\Delta p_2 = [\Delta x'_2, \Delta y'_2, -\frac{x'_2 \Delta x'_2 + y'_2 \Delta y'_2}{\xi}]^T$.

Then, we replace the parameters related to pixels by the parameters related to the 3D position in $\Delta P_M$. According to Equation (2.19), we have

$$a = p_1 = C_1 P$$

$$b = R(p_2 + \Delta p_2) \approx C_2(P - T) \tag{4.26}$$

$$c = T.$$
By substituting (4.26) into (4.25), taking the $L_2$ norm of $\Delta P_M$ and dividing by $|P|$ on both sides, we obtain the relative error as

$$\frac{|\Delta P_M|}{|P|} \approx \frac{1}{C_2} \cdot \frac{(P \times T) \cdot ((P - T) \times R \Delta p_2)}{|P \times T|^2}.$$  (4.27)

From Equation (4.27), we may observe that:

1. In addition to the matching error, the localization error is also determined by the stereo configuration (both the translation and the rotation), together with the position of the scene point.

2. The relative error might increase arbitrarily when $P \times T$ goes to zero, that is, when the point is on the direction of the baseline.

In order to better understand how the error varies w.r.t matching error, stereo configuration, and 3D scene points’ position, we show an example of the relative error distribution. The points of interest are confined on the plane $Z = 1m$. Assume that there is a half pixel error on both $x'_2$ and $y'_2$, $T = [0.6, 0.8, 0]^T m$, and $\theta = 0^\circ$. The distribution is shown in Figure 4.3. In order to clearly present the distribution, a heap map is also plotted underneath. From the distribution we see that the error goes up when a point is close to the translation direction or far away from the stereo system.

**Case 2: Two Hypercatadioptric Cameras Assembled on A Vertical Axis**

We calculate the discrete $\Delta P_M$ directly using Formula (4.15) for a special case. For which we assume that the baseline is $T = [0, 0, -0.711]^T m$ and other parameters are the same as described in the paracatadioptric case. The distribution of the relative errors is shown in Figure 4.4. From the distribution we see that the errors are small on the direction of $[\Delta x'_2, \Delta y'_2]^T$, while large in the opposite direction; and the errors go up for points away from the stereo system.
Figure 4.3: The distribution of $|\Delta P_M|/|P|$ for a single moving paracatadioptric camera. $\mathbf{T} = [0.6, 0.8, 0]^T m$, $\theta = 0^\circ$, and $\Delta x'_2 = \Delta y'_2 = 0.5 s_p$. The points of interest are on the plane $Z = 1 m$. The underneath layer is the corresponding heat map.

Figure 4.4: The distribution of $|\Delta P_M|/|P|$ for two fixed hypercatadioptric cameras. $\mathbf{T} = [0, 0, -0.711]^T m$, $\theta = 0^\circ$, and $\Delta x'_2 = \Delta y'_2 = 0.5 s_p$. The points of interest are on the plane $Z = 1 m$. The underneath layer is the corresponding heat map.
4.3.3 Sensitivity w.r.t Configuration Error

For a stereo system assembled with two fixed cameras, errors in the configuration parameters depend on measurements and should be reasonably small. In contrast, for stereo systems formed by a moving camera, which is geometrically equivalent to a stereo setup, the translation and rotation between two cameras are estimated by automatic calibration or read from other sensors. In such cases, the errors depend on the calibration algorithm or the accuracy of sensors, and are not negligible. Here, we investigate the localization error related to configuration errors.

Translation Error

Let the translation error be \( \Delta T = [\Delta t_x, \Delta t_y, \Delta t_z]^T \) and the caused localization error be \( \Delta P_T \).

Case 1: A Paracatadioptric Camera Moving on A Horizontal Plane

As before, we calculate \( \Delta P_T \) using partial differentiation,

\[
\Delta P_T = p_1 \left( \frac{\partial F}{\partial t_x} \Delta t_x + \frac{\partial F}{\partial t_y} \Delta t_y + \frac{\partial F}{\partial t_z} \Delta t_z \right) \\
= p_1 \frac{(\Delta T \times \mathbf{b}) \cdot (\mathbf{a} \times \mathbf{b})}{A_2}.
\]

In this case,

\[
\mathbf{a} = C_1 \mathbf{P} \\
\mathbf{b} = R \mathbf{p}_2 = C_2 (\mathbf{P} - (\mathbf{T} + \Delta \mathbf{T})) \\
\mathbf{c} = \mathbf{T} + \Delta \mathbf{T}.
\]

Substituting (4.29) into (4.28), we obtain

\[
\frac{|\Delta P_T|}{|\mathbf{P}|} = \frac{|(\mathbf{P} \times (\mathbf{T} + \Delta \mathbf{T})) \cdot ((\mathbf{P} - \mathbf{T}) \times \Delta \mathbf{T})|}{|\mathbf{P} \times (\mathbf{T} + \Delta \mathbf{T})|^2}.
\]

From Equation (4.30), we observe

1. The localization error is associated with the baseline and the point’s position but is not affected by the rotation since the omnidirectional camera is isotropic in each horizontal direction.
2. The error might approach infinity as $\mathbf{P} \times (\mathbf{T} + \Delta \mathbf{T})$ approaches zero.

As an example, the relative error distribution for the same set of points as in Section 4.3.2 is shown in Figure 4.6, using the same paracatadioptric stereo configuration. The translation error is $\Delta \mathbf{T} = [0.1, 0.08, 0.1]^T m$. Compared to the uncertainty caused by the matching error, the translation error has less effect on the 3D localization.

**Case 2: Two Hypercatadioptric Cameras Assembled on a Vertical Axis**

An example of the error distribution for the hypercatadioptric stereo system is shown in Figure 4.7. We assume that $\Delta \mathbf{T} = [0, 0, -0.1]^T m$ and other parameters are the same as defined in the previous hypercatadioptric case. From the distribution we see that the relative errors are the same everywhere except the center of the system. The reason is illustrated in Figure 4.5.

As the line $\overline{O_2 p_2}$ is parallel to the line $\overline{O'_2 p'_2}$. Hence,

$$\frac{|\Delta \mathbf{P}_T|}{|\mathbf{P}|} = \frac{|\mathbf{P}'|}{|O_1 \mathbf{P}|} = \frac{|O'_2 \mathbf{O}_2|}{|O_2 \mathbf{O}_1|}.$$  \hfill (4.31)

![Figure 4.5: A reconstruction with a translation error. The red lines are those with errors.](image-url)
Figure 4.6: The distribution of $\frac{|\Delta P_T|}{|P|}$ for a single moving paracatadioptric camera. $T = [0.6, 0.8, 0]^T m$, $\theta = 0^\circ$, and $\Delta T = [0.1, 0.08, 0.1]^T m$. The points of interest are on the plane $Z = 1m$. The underneath layer is the corresponding heat map.

Figure 4.7: The distribution of $\frac{|\Delta P_T|}{|P|}$ for two fixed hypercatadioptric cameras. $T = [0, 0, -0.711]^T m$, $\theta = 0^\circ$, and $\Delta T = [0, 0, -0.1]^T m$. The points of interest are on the plane $Z = 1m$. 

55
**Yaw Error**

Let the yaw error be $\Delta \theta$ and the caused localization error be $\Delta P_Y$.

**Case 1: A Paracatadioptric Camera Moving on A Horizontal Plane**

\[
\Delta P_Y = p_1 \frac{\partial F}{\partial \theta} \Delta \theta
\]

\[
= p_1(\frac{1}{A_2}[(c \times \frac{\partial b}{\partial \theta} \Delta \theta) \cdot (a \times b) + (c \times b) \cdot (a \times \frac{\partial b}{\partial \theta} \Delta \theta)]
- \frac{2A_1}{A_2}[(a \times \frac{\partial b}{\partial \theta} \Delta \theta) \cdot (a \times b)])
\]

In which,

\[
\frac{\partial b}{\partial \theta} \Delta \theta = \frac{\partial R}{\partial \theta} = C_2 \Delta R^*(P - T)
\]

\[
a = C_1P
\]

\[
b = R_{\theta+\Delta \theta}p_2 = C_2(P - T)
\]

\[
c = T
\]

\[
\Delta R^* = \begin{bmatrix}
0 & \Delta \theta & 0 \\
-\Delta \theta & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

Hence, we get the relative localization error as

\[
\frac{|\Delta P_Y|}{|P|} = \left|\frac{(P \times T) \cdot ((P - T) \times \Delta R^*(P - T))}{|P \times T|^2}\right|
\]

In this case, although the error of rotation has an effect on the localization, $\Delta P_Y$ is unrelated to the rotation itself since the omnidirectional camera is isotropic in each horizontal direction. An example of the relative error distribution is given in Figure 4.8, with $\Delta \theta = 0.1^\circ$. The errors increase for points near the translation vector or as points move away from the system.

**Case 2: Two Hypercatadioptric Cameras Assembled on A Vertical Axis**

An example of the error distribution in this case is shown in Figure 4.9. We assume that $\Delta \theta = 0.1^\circ$ and other parameters are the same as given in the previous hypercatadioptric cases. The errors are small for the points near the system and increase as points move further from the stereo system.
Figure 4.8: The distribution of $\frac{|\Delta P_Y|}{|P|}$ for a single moving paracatadioptric camera. $T = [0.6, 0.8, 0]^T m$, $\theta = 0^o$, and $\Delta \theta = 0.1^o$. The points of interest are on the plane $Z = 1m$. The underneath layer is the corresponding heat map.

Figure 4.9: The distribution of $\frac{|\Delta P_Y|}{|P|}$ for two fixed hypercatadioptric cameras. $T = [0, 0, -0.711]^T m$, $\theta = 0^o$, and $\Delta \theta = 0.1^o$. The points of interest are on the plane $Z = 1m$. 
4.3.4 Sensitivity w.r.t Calibration Error

Calibration error comes from the camera calibration algorithm. Here, we investigate the localization errors $\Delta \mathbf{P}_K$, $\Delta \mathbf{P}_P$, $\Delta \mathbf{P}_A$ and $\Delta \mathbf{P}_D$ that result from disturbances in the scaling factors, the shift of the principle point, the misalignment between the mirror and the lens, and the distortion of images respectively. Due to the complexity of derivation and the difficulty of visualizing the relationship, we calculate these localization errors using the discrete representation.

Considering the applications of the our vision system, we inspect the distributions of the relative localization errors in three hypercatadioptric omnistereo setups.

**Case 1: A Hypercatadioptric Camera Moving on A Horizontal Plane**

This setup includes a single moving hypercatadioptric camera with the following properties: the translation is $\mathbf{T} = [0.6, 0.8, 0]^T \text{m}$, no rotation angle, the scaling factors are $\alpha = 1198.8$, $\beta = 1203.3$, the principle point is at $[629.49, 503.62]$. According to the order of calibration error, we assume that the error of the scaling factor is 2%, namely, $\Delta \alpha = 23.98$, $\Delta \beta = -24.07$, the error on the principle point is $[1.5, 1.5]$ pixels, the misalignment $\Delta T_{mc} = [0.1, 0.1, 0] \text{mm}$, and the distortion factors are $\Delta k_1 = 0.05$, $\Delta k_2 = 0.005$. As previous, the points of interest are confined on the plane $Z = 1 \text{m}$. The distributions of the relative errors $|\Delta \mathbf{P}_K|/|\mathbf{P}|$, $|\Delta \mathbf{P}_P|/|\mathbf{P}|$, $|\Delta \mathbf{P}_A|/|\mathbf{P}|$, and $|\Delta \mathbf{P}_D|/|\mathbf{P}|$ are plotted in Figure 4.10, Figure 4.11, Figure 4.12, and Figure 4.13 respectively.

Observing these error distributions, we find out that the sensitivity is high for points near the translation direction and low for points perpendicular to the translation, as found previously.

**Case 2: A Hypercatadioptric Camera Moving on A Vertical Axis**

This setup includes a single hypercatadioptric camera that is assumed to move vertically along the $Z$ axis. The translation is $\mathbf{T} = [0, 0, -0.711]^T \text{m}$, and no rotation. The intrinsic parameters and the noise are the same as those in case 1. The relative error distributions are shown in Figure 4.14 ~ Figure 4.17.

From these error distributions we see that miscalibration affects little on range...
estimation when using a single moving hypercatadioptric camera. Relative errors are roughly bounded by 2%.

Case 3: Two Fixed Hypercatadioptric Cameras

This case simulates the real omnistereo configuration in our ASV. Two hypercatadioptric cameras are assembled vertically, along the $Z$ axis. The translation between them is $\mathbf{T} = [0, 0, -0.711]^{T} m$ and no rotation. The first camera has the same intrinsic parameters as the previous cases, the second camera is with $\alpha_2 = 1149.6$, $\beta_2 = 1146.1$, $u_{02} = 646.90$, $v_{02} = 515.88$. We assume that $\Delta \alpha_2 = -22.9923$, $\Delta \beta_2 = -22.9218$, $\Delta u_{02} = -1.5$ pixels, $\Delta v_{02} = -1.5$ pixels, $\Delta T_{mc} = [0.05, 0.05, 0] mm$, and $\Delta k_{12} = 0.04$, $\Delta k_{22} = 0.004$. The distributions of localization errors are shown in Figure 4.18 ∼ Figure 4.21.

From these error distributions we see that miscalibration affects a lot on range estimation when using two hypercatadioptric cameras. Relative errors are even greater than 100% in a large portion of areas.
Figure 4.10: The distribution of $\frac{|\Delta P_K|}{|P|}$ for a single moving hypercatadioptric camera. $T = [0.6, 0.8, 0]^T m$, $\theta = 0^\circ$, $\alpha = 1198.8$, $\beta = 1203.3$, $[u_0, v_0] = [629.49, 503.62]$, and $\Delta \alpha = 23.98$, $\Delta \beta = -24.07$. The points of interest are on the plane $Z = 1m$.

Figure 4.11: The distribution of $\frac{|\Delta P_P|}{|P|}$ for a single moving hypercatadioptric camera. $T = [0.6, 0.8, 0]^T m$, $\theta = 0^\circ$, $\alpha = 1198.8$, $\beta = 1203.3$, $[u_0, v_0] = [629.49, 503.62]$, and $[\Delta u_0, \Delta v_0] = [1.5, 1.5]$. The points of interest are on the plane $Z = 1m$. 
Figure 4.12: The distribution of $\frac{|\Delta P_A|}{|P|}$ for a single moving hypercatadioptric camera. $T = [0.6, 0.8, 0]^T m, \theta = 0^\circ, \alpha = 1198.8, \beta = 1203.3, [u_0, v_0] = [629.49, 503.62]$, and $\Delta T_{mc} = [0.1, 0.1, 0]mm$. The points of interest are on the plane $Z = 1m$.

Figure 4.13: The distribution of $\frac{|\Delta P_D|}{|P|}$ for a single moving hypercatadioptric camera. $T = [0.6, 0.8, 0]^T m, \theta = 0^\circ, \alpha = 1198.8, \beta = 1203.3, [u_0, v_0] = [629.49, 503.62]$, and $\Delta k_1 = 0.05, \Delta k_2 = 0.005$. The points of interest are on the plane $Z = 1m$. 
Figure 4.14: The distribution of $|\Delta P_{PK}|/|P|$ for a single moving hypercatadioptric camera. $\mathbf{T} = [0, 0, -0.711]^T m$, $\theta = 0^\circ$, $\alpha = 1198.8$, $\beta = 1203.3$, $[u_0, v_0] = [629.49, 503.62]$, and $\Delta \alpha = 23.98$, $\Delta \beta = -24.07$. The points of interest are on the plane $Z = 1m$. The underneath layer is the corresponding heat map.

Figure 4.15: The distribution of $|\Delta P_{PK}|/|P|$ for a single moving hypercatadioptric camera. $\mathbf{T} = [0, 0, -0.711]^T m$, $\theta = 0^\circ$, $\alpha = 1198.8$, $\beta = 1203.3$, $[u_0, v_0] = [629.49, 503.62]$, and $[\Delta u_0, \Delta v_0] = [1.5, 1.5]$. The points of interest are on the plane $Z = 1m$. The underneath layer is the corresponding heat map.
Figure 4.16: The distribution of $\frac{\Delta|P_A|}{|P|}$ for a single moving hypercatadioptric camera. $T = [0, 0, -0.711]^T m$, $\theta = 0^\circ$, $\alpha = 1198.8$, $\beta = 1203.3$, $[u_0, v_0] = [629.49, 503.62]$, and $\Delta T_{mc} = [0.1, 0.1, 0] mm$. The points of interest are on the plane $Z = 1m$. The underneath layer is the corresponding heat map.

Figure 4.17: The distribution of $\frac{\Delta|P_D|}{|P|}$ for a single moving hypercatadioptric camera. $T = [0, 0, -0.711]^T m$, $\theta = 0^\circ$, $\alpha = 1198.8$, $\beta = 1203.3$, $[u_0, v_0] = [629.49, 503.62]$, and $\Delta k_1 = 0.05$, $\Delta k_2 = 0.005$. The points of interest are on the plane $Z = 1m$. 

63
Figure 4.18: The distribution of $\frac{\Delta P_K}{|P|}$ for two fixed hypercatadioptric cameras. $T = [0.6, 0.8, 0]^T m, \theta = 0^\circ$. $\alpha_1 = 1198.8, \beta_1 = 1203.3, [u_{01}, v_{01}] = [629.49, 503.62], \alpha_2 = 1149.6, \beta_2 = 1146.1, [u_{02}, v_{02}] = [646.90, 515.88]$, and $\Delta \alpha_1 = -22.9923, \Delta \beta_1 = -22.9218, \Delta \alpha_2 = -29.6, \Delta \beta_2 = 29.5$. The points of interest are on the plane $Z = 1m$. The underneath layer is the corresponding heat map.

Figure 4.19: The distribution of $\frac{\Delta P_P}{|P|}$ for two fixed hypercatadioptric cameras. $T = [0.6, 0.8, 0]^T m, \theta = 0^\circ$. $\alpha_1 = 1198.8, \beta_1 = 1203.3, [u_{01}, v_{01}] = [629.49, 503.62], \alpha_2 = 1149.6, \beta_2 = 1146.1, [u_{02}, v_{02}] = [646.90, 515.88]$, and $[\Delta u_{01}, \Delta v_{01}] = [1.5, 1.5], [\Delta u_{02}, \Delta v_{02}] = [-1.5, -1.5]$. The points of interest are on the plane $Z = 1m$. The underneath layer is the corresponding heat map.
Figure 4.20: The distribution of $|\Delta P_A|/|P|$ for two fixed hypercatadioptric cameras. $T = [0.6, 0.8, 0]^T m$, $\theta = 0^\circ$. $\alpha_1 = 1198.8$, $\beta_1 = 1203.3$, $[u_{01}, v_{01}] = [629.49, 503.62]$, $\alpha_2 = 1149.6$, $\beta_2 = 1146.1$, $[u_{02}, v_{02}] = [646.90, 515.88]$, and $\Delta T_{mc1} = [0.1, 0.1, 0] mm$, $\Delta T_{mc2} = [0.05, 0.05, 0] mm$. The points of interest are on the plane $Z = 1m$. The underneath layer is the corresponding heat map.

Figure 4.21: The distribution of $|\Delta P_D|/|P|$ for two fixed hypercatadioptric cameras. $T = [0.6, 0.8, 0]^T m$, $\theta = 0^\circ$. $\alpha_1 = 1198.8$, $\beta_1 = 1203.3$, $[u_{01}, v_{01}] = [629.49, 503.62]$, $\alpha_2 = 1149.6$, $\beta_2 = 1146.1$, $[u_{02}, v_{02}] = [646.90, 515.88]$, and $\Delta k_{11} = 0.05$, $\Delta k_{21} = 0.005$, $\Delta k_{12} = 0.04$, $\Delta k_{22} = 0.004$. The points of interest are on the plane $Z = 1m$. 

65
4.4 Summary

Examining the localization error distributions, we come to the following conclusions.

1. When a single camera moves on the water surface, or virtually along a vertical axis, localization errors increase for a point close to the translation or far away from the stereo system. This is consistent with the observability of the omnistereo system studied by Xu et al. [84]. Considering the disparity distribution, we can say that the points with larger disparities have more accurate range estimation. However, if an object is too close to the stereo system, the perspective deformations are large in two images and thus accurate correspondence matching becomes difficult.

2. The 3D localization degenerates when $|P \times T|$ approaches zero. That is, a singularity ensues when a scene point is in the direction of the translation.

3. From the error distribution examples, we see that the accuracy of the matching and the rotation measurements are more critical than the accuracy of the baseline measurement.

4. Calibration errors have small effect on localization when using a single moving camera, no matter how it moves, either on the watersurface or along a vertical line. However, calibration errors result in dramatic, large errors of localization when we use two cameras.

The performance analysis provides two main rules which are important for our outdoor mapping application.

1. When using a single moving camera, the ASV should be navigated parallel to the shoreline in order to obtain the most accurate mapping results.

2. Due to noise in intrinsic parameters, range estimation using a single camera is much more accurate than the result attained using two cameras.
Chapter 5

Omnidirectional Image Processing

Omnidirectional image processing in this context refers to the procedures mainly working on images with only a loose concern with the projection models of the camera system. In our vision system, a four-step procedure is designed: feature detection, feature tracking, motion detection, and outlier rejection. The former two steps obtain pairwise correspondences in images extract metric information using stereopsis. The latter steps are to identify moving obstacles and screen robust stationary obstacles.

5.1 Feature Detection

One of the most obvious characteristics in omnidirectional images is the distortion produced by the optical system. If this type of distortion could be somehow ignored, then omnidirectional images can be tackled by the traditional image processing methods which are extensively applied in perspective images. It is noted that such distortion decreases with the distance of an object from the camera. The scenes in this work are captured in wide open areas generally away from the cameras, and hence their distortions are relatively small.

There are two basic approaches to process traditional images: dense-based [3] [83] and feature-based [63] [73]. Dense-based methods take account of each pixel in images, while feature-based methods only extract informative portions such as lines, patterns or special points and therefore are comparatively efficient. In or-
der to achieve the goal of implementing a real-time vision system, we choose the feature-based approach. Instead of detecting the shoreline [25], we select any pointwise feature that has a centered neighbor window within which the intensity values are in high contrast. The advantage of pointwise features is that they are easier to track than linewise or other type of features. Moreover, it is straightforward to extract metric information from pointwise corresponding features. On the other hand, it has a disadvantage: the selected points might be on corners of man-made objects, or branches of trees, or even waves of the water surface, and thus noisy. Extra work therefore is required to identify the features on real obstacles.

Pointwise feature detection itself is, by and large, a solved problem. Features can be extracted using the Harris corner detection method [29], the “Good feature to track” algorithm [63], or some other method. The one adopted in this work is the “Good feature to track” algorithm that selects features according to the below criterion.

Let $I$ be an image, $x = [u, v]^T$ be a pixel, and $I_u, I_v$ be the first order derivatives of $I$ about $u$ and $v$ respectively. The matrix $H$ of pixels inside a feature window $w$ is defined as:

$$H = \begin{bmatrix} \sum_{x \in w} I_u I_u & \sum_{x \in w} I_u I_v \\ \sum_{x \in w} I_u I_v & \sum_{x \in w} I_v I_v \end{bmatrix}. \quad (5.1)$$

Then, the center of the feature window is selected as a good feature if $\min(\lambda_1, \lambda_2) > t$, where $\lambda_1, \lambda_2$ are the two eigenvalues of $H$, and $t$ is a threshold value.

In this algorithm some parameters are manually tuned and determined by experimentation. These parameters and their values set in our experiments are listed in Table 5.1. It takes about 0.15 seconds to select 600 features in an omnidirectional image in the resolution of $1200 \times 1280$. Examples of feature detection are shown in Figure 5.1.

<table>
<thead>
<tr>
<th>The Parameters in Feature Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature detection window size</td>
</tr>
<tr>
<td>Threshold value of the minimum eigenvalue</td>
</tr>
<tr>
<td>The number of features</td>
</tr>
</tbody>
</table>

Table 5.1: Feature Detection Parameters
5.2 Feature Tracking

Feature tracking finds correspondences in an image sequence when a set of features have extracted in the first frame. There are two popular feature tracking methods. One is Lucas-Kanade Feature Tracker (KLT) [40], and the other is Scale-Invariant Feature Transform (SIFT) [39]. The advantage of SIFT is that it works better when features are affinely transformed. Despite this, KLT is more efficient and works well if only translation is considered, and therefore is adopted in our vision system. A brief introduction of KLT follows.

Given two images $I$ and $J$, and a pixel $x = [u, v]^T$, feature tracking finds the affine transformation parameter $A$ and the translation $d$ such that $J(x) = I(Ax + d)$, where $A$ is a $2 \times 2$ coefficient matrix and $d$ is a $2 \times 1$ vector. In our experiments, due to the size of scenes captured in images, we only consider the features’ translation. Hence, it is simplified as finding $d$ for pixels in a feature window $w$ by minimizing a matching residual $R$ defined as

$$R = \sum_{x \in w} [I(x + d) - J(x)]^2 \approx \sum_{x \in w} [I(x) + I_x(x)d - J(x)]^2$$  \hspace{1cm} (5.2)$$

where $I_x = [I_u, I_v]$.

To estimate the $d$ that minimizes $R$, we take the first order of $R$ about $d$ and set it to be zero,

$$0 = \frac{\partial E}{\partial d} \approx 2 \sum_{x \in w} I_x^T(x)[I(x) + I_x(x)d - J(x)].$$ \hspace{1cm} (5.3)$$

Solving for $d$ we get

$$d \approx \frac{\sum_{x \in w} I_x^T(x)[J(x) - I(x)]}{\sum_{x \in w} I_x^T(x)I_x(x)}.$$ \hspace{1cm} (5.4)$$

The translation $d$ is then computed using the Newton-Raphson method starting at $d_0 = [0, 0]$. This method iteratively calculates $d_{k+1}$ as

$$d_{k+1} = d_k + \frac{\sum_{x \in w} I_x^T(x + d_k)[J(x) - I(x + d_k)]}{\sum_{x \in w} I_x^T(x + d_k)I_x(x + d_k)}.$$ \hspace{1cm} (5.5)$$

The iteration stops when the relative error $|d_{k+1} - d_k|/|d_{k+1}|$ is smaller than an experimentally determined threshold, like 1% in our application.
Figure 5.1: Omnidirectional images with features marked as red points. The images in the first row are taken in Claytor Lake in Winter. These in the second row are taken in Claytor Lake in Spring. The bottom images are taken in Panama City, Florida.
In the Newton-Raphson method, the initial values should be close to the true values. Hence, the image pyramids [1] are applied. In a lower resolution level, the initial point at $d_0 = [0, 0]$ is closer to the translated point. The level of image pyramids, together with the size of feature window and the number of iterations are experimentally determined. Their values are listed in Table 5.2. It takes about 0.12 seconds to track 600 features in a $1200 \times 1280$ image.

<table>
<thead>
<tr>
<th>The Parameters in Feature Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature tracking window size</td>
</tr>
<tr>
<td>The pyramid level</td>
</tr>
<tr>
<td>The number of iterations</td>
</tr>
</tbody>
</table>

Table 5.2: Feature Tracking Parameters

5.3 Outlier Rejection

The motivation of outlier rejection is removing unreliable feature points so that only the features on real stationary obstacles are kept. It is a critical procedure because real application data usually are full of noise. In our application such noise is caused by: moving features, poorly tracked features, features on the water surface, and features with inaccurate range estimation results. In this section, we present a two-step outlier rejection method that is composed of

- A matching residual based outlier rejector, which is able to remove poorly tracked features and most of the features on water surface.

- A N-ocular stereo constraint based outlier rejector, which rejects features whose range estimation results are inaccurate.

5.3.1 Matching Residual Based Outlier Rejection

This filter is designed to remove undesired features according to their matching residuals. Matching residual in tracking algorithms is a standard measure of how
similar a feature and its correspondence are. Therefore, its use is expected to filter out poorly tracked features, and is fortunately also effective in removing features on bright specular reflections on the water surface. In the following, I will state the reasons of choosing this method and go through the details.

**Observation of Undesired Features**

By observing the tracking results, it is noted that the following phenomena result in most failures:

1. **Change of illumination.** Sometimes the lighting condition changes from one viewpoint to another, often because of shadows. Consequently, the intensity values of a feature in two images differ greatly.

2. **Affine distortion.** This happens when an obstacle is too close to the camera, which occurs when the ASV navigates near the shore or when a boat is passing by. The affine distortion of such objects is great between two images and cannot be simply modeled as translation distortion.

3. **Limited resolution.** This happens when objects are far away from the camera. Even if the objects on the shore are detected as good feature points, they cannot be tracked accurately due to the homogeneities in color and texture.

4. **Occlusion.** A common scenario in the environments where the ASV navigates is that there are lots of trees on the shore. A number of features are detected on leaves or branches, which are prone to be occluded by neighbor branches when the second image is taken from another viewpoint. As a result, these features are shifted to the neighbor branches that are visually similar in tracking, causing localization errors.

5. **Specular reflection.** This is a scenario specific to the ASV’s environments, as compared to the environments where ground or air vehicles navigate. Typically a bright and strong specular reflection of the sun is formed on the water surface in a specific direction, along with dark specular reflections are formed in other directions. In our algorithm, a large number of features are extracted
on the marginal specular reflection areas. These features on the water surface are not relevant to the obstacle avoidance task.

6. Waves. The other problematic set of features on the water surface are those on waves. They are not relevant, and in addition cannot be tracked well due to their motion and texture similarity.

A way to reduce these undesired features is to extend the KLT so that it is capable of handling the change of illumination and the affine transformation. But this improvement does not help with the remaining phenomena which take a great percentage of the entire feature population. Alternatively, in order to effectively reduce all kinds of poor features, we concentrate on the procedure of rejecting outliers, through which most undesired features are expected to be left out.

**Observation of Matching Residual**

Matching residual, as defined in Equation (5.2), measures the similarity between a feature and its tracked correspondence, and ideally equals zero. In practice, it is disturbed by noise. We thereby assume that the intensity for each pixel in a tracking window in the first image is equal to the intensity of its correspondence in the second image but disturbed by a noise factor from the standard Gaussian distribution. That is,

$$I(x + d) - J(x) \sim N(0, 1) \quad (5.6)$$

According to a property of the normal distribution, named the sum of $n$ squared independent normal random variables is a $\chi^2$ distribution with $n$ degrees of freedom, the matching residual $R$ is a $\chi^2$ distribution with $n_x \times n_y$ degrees of freedom, where $n_x \times n_y$ is the size of the tracking window. As the number of degrees of freedom increases [15], the $\chi^2$ distribution approaches a Gaussian distribution.

The assumption made above is reasonable if the matching residuals are caused by the feature tracking algorithm itself, as verified in Figure 5.3 (d), (e). However, the Gaussian noise is affected by image content. In other words, if a few features are of a certain phenomenon, such as the bright specular reflection or the motion of obstacles, their matching residuals deviate from the standard normal distribution, and hence this set of features is distinguishable from others. An illustrative exam-
ple is shown in Figure 5.3(f), in which the distribution of the matching residual for features near the bright specular reflection corresponds to the second mode in the histogram.

**Robust Estimator**

For the rejection of poorly tracked features, A. Fusiello *et al.* [15] adopt the $X^{84}$ rule to reject outliers and hence improve the feature tracking algorithm. The $X^{84}$ rule rejects the feature $i$ if

$$R_i - med_j \{R_j\} > 5.2MAD$$  \hspace{1cm} (5.7)

where $j = 1, ..., n$, $n$ is the number of features. $med$ is the median and $MAD$ is the Median Absolute Deviation which is defined as

$$MAD = med_k \{|R_k - med_j \{R_j\}|\} \hspace{1cm} (j, k = 1, ..., n)$$  \hspace{1cm} (5.8)

This rejection rule has a breakdown point of 50%. It is able to reject outliers only if they take no more than half of the entire data set. In addition, $X^{84}$ works only when the distribution is a single mode Gaussian distribution. Considering the distribution of the matching residuals in our scenarios, which may have multiple modes and more than 50% outliers, $X^{84}$ is not the best way to reject tracking outliers.

Most of the traditional robust estimators assume that the outliers occupy the minority of the entire data set. Some recent estimators claim to have a tolerance greater than 50%, but require the user provide an initial cutoff value or tune parameters, such as RANSAC [13] and RESC [86].

In order to automatically reject outliers which are more than 50% and might be from multiple distributions, we adopt the Two-Step Scale estimator (TSSE) which is proposed by Hanzi Wang, *et al.* [76]. The basic idea is to truncate the data set at the first valley of the distribution and then apply a robust scale estimator such as $MAD$ to the remaining data to reject outliers. The procedure is as follows.

1. Estimate the density of matching residuals through the kernel density estimator
\[ \hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) \] (5.9)

where \( n \) is the number of features, \( h \) is the window radius, and \( K \) is the Epanechnikov kernel function.

\[
K_e(x) = \begin{cases} 
\frac{3}{4}(1-x^2) & \text{if } x^2 < 1 \\
0 & \text{otherwise} 
\end{cases} \tag{5.10}
\]

2. Find the first valley \( v \) of the distribution by using a mean-shift algorithm. Let \( v_0 = 0 \) as the initial location, the location of the valley is found by iteratively updating \( v_k \)

\[ v_{k+1} = v_k + \alpha \cdot MV_h(x) \tag{5.11} \]

where \( \alpha \) is the step size, \( 0 < \alpha \leq 1 \), and

\[
MV_h(x) = x - \frac{1}{n_x} \sum_{x_i \in S_h(x)} x_i \tag{5.12}
\]

where \( S_h(x) \) is the region with the radius \( h \), and \( n_x \) is the number of features falling into \( S_h(x) \).

3. For the features with the residual smaller than \( v \), apply the median scale estimator

\[
M = 1.4826\left(1 + \frac{5}{n - 1} \sqrt{\text{med}_iR_i^2}\right) \tag{5.13}
\]

where the value 1.4826 is a correction factor commonly used for Gaussian approximations.

4. Reject feature \( i \) if \( R_i > M \).

### 5.3.2 N-ocular Stereo Based Outlier Rejection

The outlier rejection based on matching residuals identifies unfavorable features using visual information. However, it is unable to detect a feature whose matching residual is low but the tracked correspondence deviates from the real one. In order to identify this kind of feature, the N-ocular stereo constraint [66] is employed
which validates the tracking results from the geometry and filters the features with unreliable range estimation results affected by noise in the omnistereo system.

Viewing the omnidirectional camera as a bearing sensor, the trinocular omnistereo constraint is simplified and illustrated in Figure 5.2. Given a feature in the first image frame, if it is tracked accurately and there is no system noise, the 3D points recovered from two pairs of stereos should be overlaid at the same position, as the point A shown in the figures. However, if there is noise in the feature tracking either between image 1 and 2 or image 2 and 3, or noise in the omnistereo configuration, then the resulted positions may differ from each other dramatically. The more accurate the tracking and the system configuration are, the more consistent the estimated range can be. This implies that the N-ocular stereo can validate the correctness of tracking results and filter the features with unreliable range estimation results. So, in our 3D localization, we use four-ocular stereo to reject all features whose relative range estimation error is greater than 10%.

![Figure 5.2: (a) The trinocular geometry with tracking errors. (b) The trinocular geometry with baseline errors.](image)

5.3.3 Experimental Results

In order to show how well the two-step outlier rejection method works, we present two sets of experimental results.
Figure 5.3 demonstrates the features detected in typical images, the histograms of matching residuals, the portions of the features rejected in the first step, and the features remaining in the latter. The outlier rejection method removes most of the features on tree branches, the features on the specular reflection, and those far away from the ASV.

Figure 5.4 shows the efficiency of the outlier rejection method in two long-run experiments by overlaying the recovered locations of features on the ground truth topomaps. One of the experiments was conducted in Claytor Lake, Virginia, and the other was in Panama City, Florida. We test the algorithm for three cases: the first one is without any rejection method, the second case only uses the matching residual-based outlier rejection, and the last one using the two-step method. The total number of features in each case are listed in Table 5.3. In both experiments, about 80% features are removed.

Inspecting the remaining features that are detected as stationary obstacles, about 7% features are misdetected. Some of them are caused from features on water surface, and others are resulted when the ASV rotates.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Claytor Lake in VA</th>
<th>Bay in Panama City</th>
</tr>
</thead>
<tbody>
<tr>
<td>No outlier rejection</td>
<td>188705</td>
<td>36882</td>
</tr>
<tr>
<td>Matching residual based (percentage)</td>
<td>135907 (72.02%)</td>
<td>28916 (78.4%)</td>
</tr>
<tr>
<td>Matching residual based and N-ocular constraint (percentage)</td>
<td>42482 (22.51%)</td>
<td>7308 (19.81%)</td>
</tr>
</tbody>
</table>

Table 5.3: Results of Outlier Rejection.
Figure 5.3: The images in the first row are the results of feature detection. The second row shows the histograms of matching residuals. The images in the third row are the results of the matching residual based outlier rejection. The last row shows the results using the two-step outlier rejection method. The rejected features are marked in green.
Figure 5.4: (e) and (f) are the results of online range estimation. The experiment in the left column was tested in Claytor Lake, Virginia. The entire run is about 2km. The width of the river is from 80 to 200 meters. The experiment in the right column was tested in Panama City, Florida, in AUVFest 2007. The entire run is about 300 meters. The gray lines are the trajectory of the ASV. Red points are the results of range estimation for stationary features.
5.4 Motion Detection

Motion detection aims to identify and track the portion of images capturing independently moving objects. It is a topic that has been extensively studied [52] [40] for decades. In contrast to most motion detection algorithms, we detect motion in omnidirectional images taken by moving cameras, which causes challenges from both the moving background and noisy data. We propose a motion detection approach which employs the epipolar constraint to select moving candidates and utilizes a nearest neighbor validation to filter out false candidates.

In the first of this subsection, I briefly review the motion detection related work, point out the specific challenges in our applications and state the reasons for choosing our motion detection method. Then, I introduce the method in detail and present experimental results.

5.4.1 Related Work

The scenarios in motion detection can be classified into two categories with respect to cameras' states: stationary or moving. When a camera is stationary, the background environment is static in captured images and only dynamic objects move. For instance, in most video surveillance, cameras are fixed somewhere to monitor the behavior of people or objects coming in and out of the field of view. In the case of moving cameras, the captured background shifts, while dynamic objects move independently. The most common example is vision based robot navigation, as in our application, where cameras are mounted on the ASV and hence move along with the ASV.

Motion detection in the first category is relatively simple, since moving objects can be detected by background subtraction algorithms [28] [8]. This family of algorithms identifies moving objects from the portion of an image frame that differs significantly from a static background model, which is typically built by a Gaussian mixture model [90] and is robust against changes in illumination, high-frequency background objects, such as tree leaves, water waves, etc. and changes in the background geometry.

In the second category, motion compensation [9] of backgrounds is often applied
when the camera’s motion can be estimated and the background can be viewed as a planar plane, for example, the ground plane. With the assumption of planarity, a motion model of the background is estimated and employed to identify independently moving objects whose motions are inconsistent with the model.

Both background subtraction and motion compensation approaches are generally dense-based. Beyond that, another popular way to detect motion is based on sparse optical flow [79]. A number of features are detected and tracked in image sequences and grouped into the background and independent moving objects by clustering [77] or factorization [75] [73] [82] methods. Alternatively, the sparse optical flow pattern of backgrounds [67] [35], especially the Focus-Of-Expansion (FOE) and Focus-Of-Contraction (FOC), are investigated and applied for detecting independent movements.

5.4.2 Challenges in Our Application

Although there are many motion detection methods, most of them are not suited to our task, considering the following facts:

1. **Moving backgrounds.** In our case the cameras are moving with the ASV, so that stationary environments are moving relative to the cameras. Therefore background subtraction algorithms are not suitable.

2. **No reference plane.** Although the water surface is flat and can be viewed as a plane for some scenarios, most of the features in our environment are detected on trees, man-made objects, or cluttered objects on the shore. Hence, we cannot assume that the background is planar. Even though the motion of the camera is known, it is hard to compensate the motion of the background.

3. **Varying depths.** Two stationary objects of different depths might have optical flows in opposite directions when the camera moves. Figure 5.5 shows an example, in which the optical flows of the features on a white house circled in the green ellipse are in counter-clockwise direction, while the optical flows of the features on trees behind it are in clockwise direction. The method based on the background optical flow pattern fails in this situation.
4. **Unknown number of moving obstacles.** If motion detection is formalized as a classification problem, it is necessary to know or estimate the number of classes. But in our application no prior information about the number of moving obstacles, e.g. other vessels, is available.

5. **Noisy data.** Other than features selected on stationary and moving obstacles, there are a number of features on the water surface, for instance, on waves and marginal area of bright specular reflection regions, which are moving but indeed are noise.

![Figure 5.5: An omnidirectional image marked with features and optical flows. Red points are features detected. Blue lines indicate the optical flow.](image)

**5.4.3 A Two-Step Motion Detection Algorithm**

A fortunate fact is that our system uses GPS and a gyroscope to collect the position and attitude information of the ASV at each time instant. This simplifies
the motion detection problem because the relative movement of the cameras is measured, in terms of which the epipolar constraint can be calculated and applied to choose stationary features. The issues that need to be addressed are how to robustly select moving candidates based on the epipolar constraint, and how to separate moving features from false candidates.

Epipolar Constraint Residual based Motion Detection

Following the epipolar constraint introduced in Section 2.2.3, the epipolar constraint residual is defined as

\[ R_e = (p_1^T E p_2)^2, \]  \hspace{1cm} (5.14)

where \( p_1, p_2 \) are pairwise correspondences and \( E \) is the essential matrix. \( R_e \) has a distribution analogous to the matching residual, hence we use TSSE to separate stationary features from moving candidates. Examples of \( R_e \) are shown in the second row of Figure 5.6 and Figure 5.7.

Nearest Neighbor Validation

After a feature is selected as a moving candidate, we use a simple method to validate if it is a moving feature or a outlier. Assume there is more than one feature detected on a moving object and all of features on a single moving object are close to one another. Then the optical flow of a moving feature is compared with that of its closest neighbor. The motion similarity is defined as

\[ S = \frac{v_1 \cdot v_2}{|v_1||v_2|} + \frac{|v_1|}{|v_2|} + \frac{|v_2|}{|v_1|}, \]  \hspace{1cm} (5.15)

where \( v_1 \) and \( v_2 \) are optical flow vectors of the two nearest moving features. The threshold value of the motion similarity is experimentally determined. In our case, it is set as 4. If \( S > 4 \), the feature is considered an outlier.

5.4.4 Experimental Results

Some of the experimental results are presented in Figure 5.6 and Figure 5.7. The images in the first rows of Figure 5.6 and Figure 5.7 are the feature tracking results, in which detected features are marked in red and their optical flow are
depicted by blue lines. The plots in the second rows are the histograms of epipolar constraint residuals. Moving candidates selected by TSSE are marked in green in the third rows. The results using nearest neighbor validation are shown in the last rows, in which yellow points are detected as false candidates and green points are moving features.

In Figure 5.6, most of features on two boats are detected correctly as moving obstacles, and features on the water surface are rejected by nearest neighbor validation. In Figure 5.7, although lots of mis-tracked features and features on the water surface are selected as moving candidates, most of them are rejected in the validation stage. These experiments show that this algorithm is effective.
Figure 5.6: Experimental results of motion detection.
Figure 5.7: Experimental results of motion detection.
5.5 Summary

In this chapter we introduced a four-step omnidirectional image processing framework to indentify moving obstacles and localize stationary obstacles. The detailed image processing procedure, graphically depicted in Figure 5.8, is presented here as a summary of this chapter. This procedure, being an essential part of the entire vision system, works in real-time and effectively. Its performance is validated in real applications in the next chapter.

![Diagram of vision processing]

Figure 5.8: An overview of vision processing.
Chapter 6

Experiments and Validation

This chapter introduces the experimental platform - the autonomous surface vehicle (ASV) - on which the vision system is tested, summarizes the framework of our entire vision system, and present localization results which are verified on ground truth maps.

6.1 Testbed Setup

The omnidirectional vision system is tested on an autonomous surface vehicle (ASV) shown in Figure 6.1, 6.2. The ASV, as an experimental platform, is a product of the Autonomous Systems and Controls Laboratory (ASCL) at Virginia Tech. It consists of:

- An inflatable pontoon.
- A gasoline generator.
- Two DC motors.
- A GPS.
- A gyro sensor.
- Two laptops.
- A paracatadioptric camera or two hypercatadioptric cameras.
The ASV is $2.7m \times 1.5m$ in size and weighs $125kg$. It can navigate automatically by waypoints at an average speed of $1.5m/s$. The ASV is capable of navigating almost fully automatically with the path planning algorithm implemented by Xu and Reed [56], using the map generated by the omnivision system.

### 6.2 Omnivision System

The ASV employs omnidirectional cameras to sense its environment. With the advantage of the large field of view, $360^\circ$ in horizontal and $80^\circ$ in vertical, omnidirectional cameras are able to capture views of the entire area surrounding the ASV and detect obstacles. We have two generations of omnivision systems. The first one uses a paracatadioptric camera, mounted to the front of the ASV, as shown in Figure 6.1. Stereo vision is obtained when the ASV is moving. The disadvantage of this system is the lack of capability to detect obstacles in-line with the trajectory direction, according to the performance we analyzed in Section 4. To conquer this problem, the second system is designed which mounts two hypercatadioptric cameras vertically on the ASV, as shown in Figure 6.2.

![Figure 6.1: A photo of the ASV, equipped with a paracatadioptric camera.](image1)

![Figure 6.2: A photo of the ASV, equipped with two hypercatadioptric cameras.](image2)

The implemented omnivision system captures images, detects features and recovers the 3D information of obstacles in real-time. The flowchart of vision processing is shown in Figure 6.3. Two laptops are used in the ASV. One is for the
control part that manages the mechanics of the ASV and receives GPS and gyro data about the ASV’s current status. The other is for the vision processing. Two laptops work parallelly and communicate the sensor data and the range estimation results using the stable TCP connection. The frequency of vision processing is 5 frames/second.
Figure 6.3: Overview of the omnivision system.
6.3 Experimental Results

The ASV navigates in riverine environments to build shoreline maps. Hence, our experiments are conducted in open area full of noise. Here we present some results of our vision system. The results of range estimation are overlaid on real topomaps for validation. Figures 6.4 and 6.5 show two results attained by the single paracatadioptric camera. Figure 6.6 shows a result obtained by the two hypercatadioptric cameras.

Generally speaking, the results are promising. But a small percentage of features are misdetected as obstacles. For example, in the experiment shown in Figures 6.4, about 7% features are misdetected, like the features in the middle of river shown on the right up corner of the map. Some of them are caused from reflections on the water surface, and others are resulted when the ASV rotates. In the experiment shown in Figures 6.5, about 4% features are misdetected, which are those features far away from the ASV.
Figure 6.4: A result of online range estimation using a paracatadioptric camera. The experiment was tested in Claytor Lake. The entire run is about 2km. The black line is the trajectory of the ASV. Red points are the results of range estimation for features.
Figure 6.5: A result of online range estimation using a paracatadioptric camera. The experiment was tested in Panama City, Florida. The entire run is about 300 meters. The blue line is the trajectory of the ASV. Red points are the results of range estimation for features.
Figure 6.6: A result of online range estimation using two hypercatadioptric cameras. The experiment was tested in Claytor Lake. The entire run is about 500 meters. The dark line is the trajectory of the ASV. Red points are the results of range estimation for features.
Chapter 7

Conclusions and Future Work

Motivated by the application of navigating an ASV in riverine environments, in this dissertation we have addressed the essential problems on developing an omnidirectional vision system to detect obstacles in real time. Conclusions based on this work are drawn and future directions are pointed out as follows.

7.1 Conclusions

Chapter 2 introduced the single-view and two-view geometries of omnidirectional cameras. In omnidirectional stereopsis, an explicit formula was derived to recover the 3D location given two pairwise correspondences. This formula is suitable to any camera projections and stereo configurations. It also implies that the features on the trajectory direction are unobservable.

Chapter 3 presented a method to calibrate paracatadioptric and hypercatadioptric cameras. The calibration performance was investigated numerically using Monte Carlo simulation, and provided us with a comprehensive view of the accuracy that a calibration algorithm can obtain.

Chapter 4 numerically investigated the performance of the omnistereo vision systems by evaluating the maximum detectable range and the uncertainty of 3D localization with respect to disturbances on camera calibration, stereo configuration, and image processing. Performance analysis is critical in large-scale applications, informing us how precise and stable a system can be. Performance analysis iden-
tified several important observations:

1. The points on the baseline direction are unobservable.

2. The points near the direction perpendicular to the stereo baseline can be reconstructed more accurately than these near the baseline direction.

3. Range estimation errors stemming from calibration in the case of using two omnidirectional cameras are much larger than these in the case of using a single camera.

Chapter 5 and 6 introduced the omnidirectional image processing and the framework of the entire vision system. The procedure of image processing was designed to detect and track pointwise candidate features, and further focus on outlier rejection to identify moving features and remove false candidates. This strategy made the vision system very efficient. Validated by the outdoor experiments, the system is capable of localizing stationary obstacles and detecting moving objects in real time with high accuracy.

7.2 Future Work

A common goal of autonomous navigation is to estimate the motion of the robot and reconstruct the environment, a simultaneous localization and mapping (SLAM) problem. In this work, we combined omnidirectional cameras with GPS and gyroscope, so that SLAM is simplified to a SFSM problem. Combining different sensors is popular in robotics. Hence, a direction would be to combine other sensors such as radar or laser range finders with the omnidirectional cameras to attain more precise localization results.

In this dissertation, we neglected the problem of identifying previously detected objects. A radar or a laser alone does not provide sufficient information to solve this problem [46]. But images contain rich information which can be used to register the scenes explored multiple times. This would be another potential direction for future work.
Bibliography


IEEE Transactions on Pattern Analysis and Machine Intelligence,

[53] Radu Orghidan. Catadioptric stereo based on structured light projection,

[54] Peter Peer and Franc Solina. Panoramic depth imaging: Single standard
camera approach. International Journal of Computer Vision, 47:149–160,
April-June, 2002.

[55] Simon J.D. Prince, Ke Xu, and Adrian David Cheok. Augmented reality camera
tracking with homographies. IEEE Computer Graphics and Applications,

[56] Caleb Reed. Experiments in Real-time Path Planning for Riverine Environ-


[58] Jamie N. Riggins. Location Estimation of Obstacles for an Autonomous Sur-

[59] Jeffrey J. Rodriguez and J. K. Aggarwal. Stochastic analysis of stereo quan-
tization error. IEEE Transactions On Pattern Analysis and Machine Intelli-

[60] D. Scaramuzza, A. Martinelli, and R. Siegwart. A flexible technique for acc-
curate omnidirectional camera calibration and structure from motion. IEEE

[61] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A flexible
technique for accurate omnidirectional camera calibration and structure from


