Mapping and Localization from a Panoramic Vision Sensor

Mapping and Localization from a Panoramic Vision Sensor

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam,
op gezag van de Rector Magnificus
prof. mr. P.F. van der Heijden
ten overstaan van een door het college voor promoties ingestelde
commissie, in het openbaar te verdedigen in de Aula der Universiteit
op vrijdag 14 november 2003, te 12:00 uur.

door

Roland Bunschoten

geboren te Naarden

Promotor:	Prof. dr. ir. F.C.A. Groen
Co–promotor:	Dr. ir. B.J.A. Kröse
Commissie:	Prof. dr. P.W. Adriaans
	Dr. ir. P.P. Jonker
	Prof. dr. ir. B.M. ter Haar Romeny
	Dr. J. Santos-Victor
	Prof. dr. P.J. Werkhoven

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

IAS intelligent autonomous systems

Universiteit van Amsterdam

The research described in this thesis was partly sponsored by the Japanese "Real World Computing" project and partly carried out in the European ITEA "Ambience" project.

Advanced School for Computing and Imaging

AMBIENCE

This work was carried out in graduate school ASCI.
ASCI dissertation series number 94.

ISBN 90-9017279-3

This thesis was typeset by the author using the LaTeX2$_\varepsilon$ document preparation system.
Printed and bound by Febodruk B.V., Enschede, The Netherlands.

voor Helma

List of Publications

The research described in this thesis has resulted in the following other publications:

R. Bunschoten and B. Kröse. 3D scene reconstruction from cylindrical panoramic images. *Robotics and Autonomous Systems*, 41:111–118, 2002.

R. Bunschoten and B. Kröse. Range estimation from a pair of omnidirectional images. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1174–1179, Seoul, Korea, May 2001.

R. Bunschoten and B. Kröse. Robust scene reconstruction from an omnidirectional vision system. *IEEE Transactions on Robotics and Automation*, 19(2):351–357, April, 2003.

R. Bunschoten and B. Kröse. Visual odometry from an omnidirectional vision system. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, May 2003. (to appear).

B. Kröse and R. Bunschoten. Probabilistic localization by appearance models and active vision. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2255–2260, 1999.

B. Kröse, R. Bunschoten, S. ten Hagen, B. Terwijn, and N. Vlassis. Environment modeling and localization from an omnidirectional vision system. *IEEE Robotics and Automation Magazine (special issue on "Panoramic Robotics")*, 2003. (to appear).

B. Kröse, R. Bunschoten, N. Vlassis, and Y. Motomura. Appearance based robot localization. In G. Kraetzschmar, editor, *IJCAI-99 Workshop on Adaptive Spatial Representations of Dynamic Environments*, pages 53–58, 1999.

B. Kröse, N. Vlassis, R. Bunschoten, and Y. Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391, April 2001.

Contents

Chapter 1

Introduction

1.1 Mobile Service Robots

Mobile robots have a wide applicability and they are gradually making their way into our daily lives. Autonomous floor cleaning robots are already employed in shopping centers, airports, factories and other large buildings to perform a tedious cleaning task at nighttime. The “ST82 R” robot, produced by Hefter Cleantech, currently cleans floors in five Dutch supermarkets. Mobile robots perform security and surveillance tasks such as fire detection and intruder detection. The operational “MOSRO 1” surveillance robot, developed by Robowatch Technologies, uses a radar system that can identify contours of intruders even through walls. Mobile delivery robots, developed by Helpmate Robotics Inc., distribute medicines in over 100 American hospitals. Experience with this “Helpmate” robot shows that the robot performs the distribution tasks faster and more reliably than humans do. The above examples illustrate that mobile service robots can save both time and money, can perform tasks that humans consider dangerous or tedious, and may (unintentionally) entertain us in the process.

In order for a mobile service robot to perform its navigational tasks, it should be able to determine where it is. This *localization* problem, determining the pose (position and orientation) with respect to a global map of the workspace, has occasionally been referred to as “the most fundamental problem to providing a mobile robot with autonomous capabilities”[9].

Today’s commercial service robots often rely on an a priori provided map of their workspace. Such a map contains the locations of easily recognizable artificial landmarks, which have been mounted at strategically chosen locations. Localization is such a scenario may be considered a solved problem. Ideally, a mobile robot should be capable of learning and maintaining a map of its workspace while it performs its tasks. It should be able to do so without relying on the presence of artificial landmarks. The problem of *mapping*, creating a map for localization from sensor data collected during navigation in an initially unknown environment, is currently an active research topic.

1.2 Vision for Mobile Robots

Humans have evolved to rely primarily on vision for localization and navigation tasks (and many other tasks as well). Although we certainly use our other senses as well, these provide less informative clues as to where we are. Similarly, in robotics certain external sensors — sensors that measure aspects of the external world — may yield more informative clues than others may.

Early mobile robots were usually equipped with a ring of ultrasonic range sensors from which rough estimates of distances to surrounding objects can be derived. Nowadays, robots can be equipped with laser range finders providing precise distance information. Laser range finders can be used to construct accurate 2-D (or even 3-D) geometric maps of a robot's workspace. Still, range information, however accurate it may be, may not be very informative for localization because many distinct places in a robot's workspace can have a similar geometric structure. Moreover, today's laser range finders are very expensive.

Vision sensors (i.e. cameras), on the other hand are cheap nowadays. A camera image provides abundant information about a scene observed and may thus yield stronger clues for localization. Employing a mobile robot with a camera furthermore opens up the possibility to perform a wide range of visual tasks, such as recognizing objects to grasp and recognizing people to interact with.

In spite of these advantages, range sensors are still very popular and commonly employed in studies concerning mapping and localization. This is likely because a range sensor provides both distance and bearing information, whereas a camera measures intensity of light reflected by objects and bearing only. The fundamental data association problem of establishing matches between a local sensor measurement and a (partially build) map is more complicated for a bearing only sensor because a match already established constrains other potential matches to a lesser extend. Furthermore, distance information — which is required for the construction of a metric map of the workspace — can only be recovered indirectly via triangulation of established point matches between at least two images.

In light of the above challenges, the large scope of potential applications and its cost effectiveness, in this thesis we explore the possibility of using a vision sensor for robot map building and localization.

1.3 Panoramic Vision using Mirrors

Conventional cameras have a relatively narrow field of view. In order to get an overall impression of its surrounding environment, a robot equipped with such a camera should actively look around. It could for instance use a pan-tilt-zoom mechanism to aim the camera in different directions, or it could rotate its body. It would be more practical if the robot were equipped with a panoramic vision sensor that provides a 360° impression of the surrounding environment instantaneously. One approach to achieve such panoramic

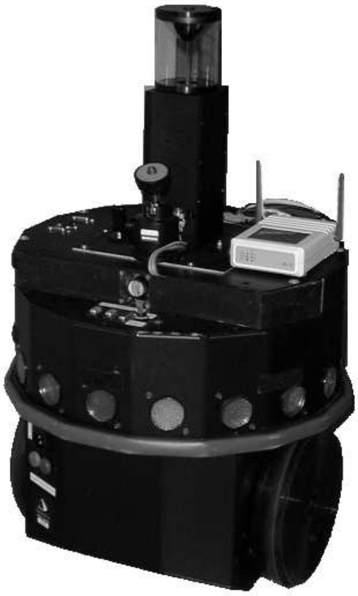


Figure 1.1: Our experimental platform. The robot is a Nomad Super Scout II, manufactured by Nomadic Technologies Inc. The panoramic vision sensor, manufactured by Accowle Inc., is situated on top of the robot.

sight is to observe the world via a curved mirror. Vision sensors that combine mirrors and lenses are called *catadioptric* vision sensors. Dioptrics is the science of refracting elements. Catoptrics is the science of reflecting surfaces, or mirrors. Their combination is therefore called catadioptrics.

Mirrors, lenses and their useful properties have long been known to mankind. The earliest known mirrors were made out of polished volcanic glass and date back to c. 7000 BC. Lenses made out of crystal rock, dating back to c. 900 – 700 BC have been found at sites in Assyria. According to the Greek writer and philosopher Plutarchus (46 – 120 AD), the famous Greek mathematician Archimedes (287 – 212 BC) constructed concave mirrors during the siege of Syracuse by the Romans (214-212 AD). Supposedly, the mirrors were used to burn down Roman ships with reflected sunlight. Although this particular story remains unsubstantiated, ancient concave burning-mirrors made from polished metal have been found at sites in Egypt, China and Greece. One of the earliest known mathematical studies concerning the properties of mirrors and lenses was done by the Arabic mathematician, physicist and astronomer known as Alhazen (c. 965 – 1039 AD). He described results of experiments with spherical and parabolic mirrors (and many other optical phenomena) in his treatise "Kitab al-manazir". In 1270, an influential Latin translation known as "Opticae thesaurus" was published in Europe. The burning mirrors from ancient times found a new application in astronomy. In 1672, Sir Isaac Newton presented his reflective telescope, featuring a concave parabolic mirror. His revolutionary design paved the way to magnification of object far beyond what could ever be obtained with a lens.

Nowadays, the use of curved mirrors to enable panoramic sight for mobile robots is quickly gaining popularity. Although there exist alternative methods to obtain instantaneous panoramic vision (the interested reader is referred to the book edited by Benosman and Kang [2] and the proceedings of the workshops on omnidirectional vision [74, 75]), catadioptric systems appear particularly well suited for mobile robot applications. They are compact, relatively cheap, and have no moving parts so that they suffer little from wear and tear and consume little power. In this thesis, we present methods for robot mapping and localization in which we seek to take advantage of the large field of view offered by a catadioptric panoramic vision sensor. An image of our experimental platform, a Nomad Scout robot, with its catadioptric panoramic vision sensor is displayed in figure 1.1.

1.4 Original Contributions

In this thesis, we describe our research on visual mapping and localization. We have contributions in the areas of vision sensor design and calibration, robot localization, panoramic stereo vision and estimation of robot poses from images acquired during navigating in a previously unknown environment.

- We derive the parameters of a hyperboloid mirror so that the resulting catadioptric panoramic vision sensor meets a given view angle specification. We propose a simple method to calibrate the vision sensor. We show how various virtual cameras can be constructed. A virtual camera re-projects the catadioptric image onto a different surface. Throughout the thesis, we argue and demonstrate that images obtained by such virtual cameras are better suited for certain tasks than the panoramic images from which they are derived.

- We present a novel appearance-based model for probabilistic robot localization.

- We analyze the epipolar geometry for cylindrical panoramic images and propose a novel parameterization of (sinusoidal) epipolar curves that enables efficient stereo matching across multiple images obtained at (non co-linear) camera poses.

- We present a novel method to estimate a camera trajectory from a sequence of catadioptric images. Unlike many approaches presented in literature, our approach does not require many feature correspondences across many images. Instead, our method estimates the relative pose relationship between pairs of (catadioptric) images. For each pair, feature correspondences between two (virtual) cylindrical panoramic images are used to estimate the rotation and direction of translation. The length of the translation is subsequently estimated by registering two (virtual) planar perspective images of the ground plane.

1.5 Thesis Overview

Each of the contributions described in the previous section are presented in a separate chapter of this thesis.

Chapter 2 is not directly related to the problems of mapping and localization. It introduces the field of panoramic vision with a particular emphasis on vision sensor designs involving a curved mirror to obtain a panoramic field of view. We discuss the geometric properties of our particular sensor, present a calibration scheme, and show how the image obtained by the sensor can be mapped to into other valid perspective image representations.

Chapter 3 addresses the problem of mapping and localization based on images. We describe our approach, which models the relation between robot poses and the appearance of observed images directly. Modeling is done based on a set of images collected at known poses throughout the robot's workspace. Such "appearance-based modeling" was first proposed in the field of object recognition. Since then, it has appeared in a variety of contexts, including robot mapping and localization. An important feature of our method is that it builds on a probabilistic framework for robot localization. Probabilistic approaches are generally less brittle than approaches that maintain a single pose estimate because uncertainty is explicitly represented and reasoned with. Appearance-based approaches require many training images to learn the map. In the resulting map, the concept of free versus occupied space is not explicitly represented. Collision free navigation can thus not solely be based on the appearance-based model.

Chapter 4 addresses both these issues. We present an efficient method to estimate depth from multiple panoramic images obtained at different poses. We show how the estimated depth information can be used to predict the appearance of images obtained at nearby poses. In principle, the methods discussed can be used to generate training images for the method presented in chapter 3. In order to estimate depth from images, the relative poses at which the images are acquired have to be known. In the chapter we derive the required pose information from wheel odometry and refine the pose information using vision.

Chapter 5 we present a method to estimate relative camera poses using visual information only. We apply our method to reconstruct a robot trajectory from consecutive images acquired during navigation in a previously unknown workspace. Like wheel odometry, small errors accumulate so that the estimated end pose in a trajectory may be far from the true end pose. Unlike wheel odometry, visual odometry is not "blind". A previously visited place can be recognized. This provides a handle to correct the estimated past trajectory. The methods presented in chapter 3 and in chapter 4 require that the poses at which input images are obtained are known. The method presented in this chapter could be used to estimate the required pose information automatically.

Each chapter in this thesis has a section dedicated to discussion and conclusions. Chapter 6 draws general conclusions and indicates possible directions for future research.

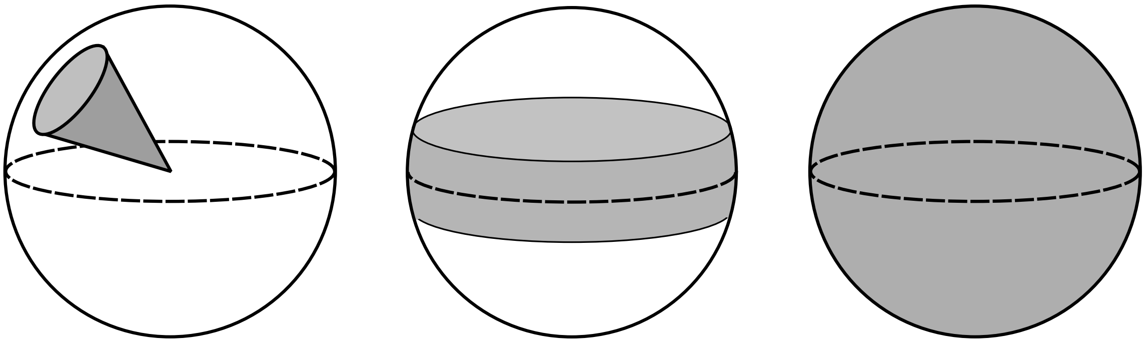
Chapter 2

Panoramic Vision

Conventional perspective cameras (film, digital or video) have a relatively narrow field of view (typically 30° – 60°). In contrast, biological eyes often have a much larger field of view. A pair of human eyes, for instance, cover a field of view of 120° in horizontal direction, and 135° in vertical direction. There even exist species — diurnal insects, nocturnal insects and some crustaceans — which have compound eyes enabling them to see all around. Panoramic vision sensors, replicating such biological panoramic vision, can be exploited in a variety of mobile robotics tasks.

A panoramic image can be a robust indicator of the robot pose and can be used for vision based robot localization [42, 26]. In many mobile robot applications images arrive sequentially as the robot moves around in its workspace. Panoramic images that are acquired at consecutive poses have almost complete visual overlap. This simplifies the matching of salient image features. Mobile robot applications that depend critically on establishing reliable feature correspondences include navigation [111, 112, 35, 26] and camera motion estimation [86, 28, 50, 113]. Furthermore, the computation of camera motion from panoramic images is more stable because small camera displacements give rise to image motion patterns distinctly different from image motion patterns caused by small rotations [19]. Finally, a 3-D scene reconstruction can be obtained from just a few images [13].

In this chapter we introduce our panoramic vision sensor, which consists of a conventional camera and a convex hyperboloid mirror mounted in front of the camera lens. In section 2.1 we review different mechanisms to acquire panoramic images. In section 2.2 we present the geometric image formation of our panoramic vision sensor. The design of the mirror and the calibration of the sensor are discussed in section 2.3. It is possible to map the images acquired by the sensor to other surfaces. This property can be used to construct virtual cameras. In section 2.4 we show how a spherical image, a cylindrical image and a planar perspective image can be derived from a panoramic image captured by the mirror based sensor. Such images can be regarded as though they are acquired by cameras with different projection surfaces. Conclusions are presented in section 2.5.



(a) directional (b) panoramic (c) omni-directional

Figure 2.1: The view sphere and camera classes. (a) The field of view of *directional* cameras is a subset of a hemisphere of the view sphere. (b) The field of view of a *panoramic* camera covers at least one great circle on the view sphere. (c) *Omnidirectional* cameras can see in all directions.

2.1 Panoramic Vision Sensors

Conventional perspective cameras measure the intensity of rays that all pass through a single point in space. This point is known as the *effective pinhole*, or the *viewpoint* of the camera. A camera that measures irradiance from a single viewpoint is called a *central* camera. The geometric image formation of a conventional camera can be described by the pinhole and by the plane onto which the scene is projected. The field of view of a camera can be characterized by the area that is covered on the surface of a sphere centered at the viewpoint. This sphere is called the *view-sphere*. Figure 2.1a displays the view-sphere and the field of view of a conventional camera. The field of view of a conventional camera is always less than a hemisphere (half the view sphere). Conventional cameras are called *directional* because they face a particular direction, which is described by the normal to the plane onto which the environment is projected. It is possible to construct (virtual) cameras that perform a projection onto a surface other than a plane. When the field of view of such a camera covers at least one great circle[1] on the view sphere, the camera is called *panoramic*. Panoramic cameras provide a 360° field of view in one direction, whereas in other directions their field of view is limited. An illustration of the field of view covered by a panoramic camera is shown in figure 2.1b. When a camera can see in all directions it is called *omnidirectional*. The field of view covered by an omnidirectional camera covers the entire view sphere, which is illustrated in figure 2.1c.

Rotating cameras. The first approaches to obtaining panoramic images were software based. In 1988 Zheng and Tsuji [115] were the first to construct a *digital* panoramic image. In their approach, a sequence of images was acquired by panning a camera. Their panorama was constructed by taking a single column from each successive image and

[1] A great circle on a sphere is a circle that has the same radius as the sphere.

concatenating them. This approach is known as the slit camera approach. Later, it was realized that perspective images obtained by a camera rotating about its center of projection are related by a parametric model (planar homography, see chapter 5). As a result the images can be registered by fitting a small number of unknown parameter values, and composed into a panoramic image [87]. A similar method can be applied when the scene observed is known to be planar. In recent years there has been a growing interest in registering images acquired by a freely moving camera [81].

Dedicated hardware. Software based approaches using a rotating camera can produce high resolution panoramic images. The main disadvantage is that it is time consuming to acquire a panoramic image. The use of these methods is limited to static scenes. Today, there exist a wide range of vision sensors that are specifically designed to capture a panoramic image instantaneously. Sensors consisting of multiple synchronized directional cameras, each of them facing a distinct direction, can deliver high resolution omnidirectional or panoramic images in real-time. In [11] a design involving 5 cameras attached on a disc is presented that captures panoramic images in real-time. Immersive Media [59] produces a panoramic vision sensor consisting of a cluster of video sensors arranged in a compact dodecahedral framework. Viewplus [101] produces an omnidirectional vision sensor that combines twenty stereo vision units arranged on the faces of a icosahedron. The panoramic vision system produced by Fullview [24] uses four cameras, each observing a planar mirror. The mirrors are arranged in an inverted pyramid.

Cameras and curved mirrors. It is well known that curved mirrors can be used to increase an otherwise limited field of view. In what follows, we will focus on catadioptric sensors combining a single lens with a single curved mirror. To obtain a wide field of view, one generally uses a mirror whose shape is a convex surface of revolution of a curve describing the mirror profile. The mirror surface is thus determined by the profile curve. The shape of the mirror determines the direction in which rays originating in the camera pinhole are reflected. In [34] a family of curved mirror shapes is derived that reflect a world plane below the mirror in such a way that the image of the plane appears undistorted in the catadioptric image. A planar mirror could be used for this purpose, but planar mirrors do not increase the field of view. Using their curved mirror, an extremely wide field of view can be obtained. In [6] a family of mirror shapes is studied that yield a linear relationship between the angle under which a ray enters the mirror and the angle of reflection onto the camera.

We are particularly interested in mirror shapes that can be used to construct panoramic vision sensors that capture light rays that would meet at a single point in space had they not been reflected by the mirror. The point where the rays would meet is called the *effective viewpoint* of the vision sensor. Figure 2.2 shows a diagram of a catadioptric vision sensor with a single effective viewpoint.

Catadioptric systems that have a single effective viewpoint have two attractive properties. First, because the sensor measures intensity of rays originating from a single point in

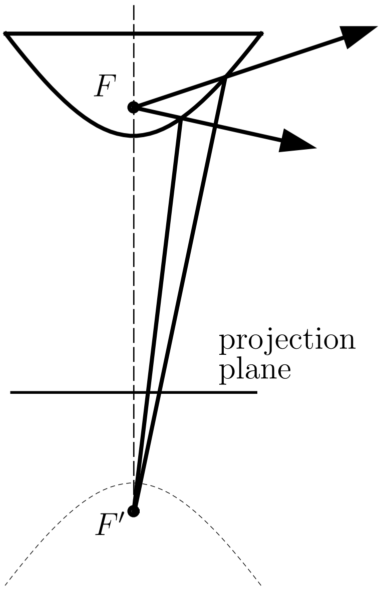


Figure 2.2: A central catadioptric systems. Rays that would have passed through the focal point of the mirror F are reflected in such a way that they pass through the focal point F' of the camera. As a result, the camera measures intensity of rays originating in F. Therefore, F is called the effective viewpoint of the camera.

space, it is possible to produce other types of geometrically correct perspective images. In section 2.4 we describe in detail how virtual central spherical, planar and cylindrical cameras can be implemented. These virtual cameras have a pinhole that coincides with the effective viewpoint of the catadioptric system. Only the surface onto which the environment is projected (a sphere, a plane and a cylinder respectively) is different. Secondly, images acquired by such a sensor from different positions can be related via the epipolar geometry. Each point in an image defines a ray. The epipolar geometry expresses the fact that the rays for corresponding points in two images meet at a single point in space. The epipolar geometry can be established for all cameras that perform a central projection. It can be used to find matching image features in two views, to estimate relative camera poses, and for 3-D reconstruction. The epipolar geometry and its applications are discussed in detail in chapter 4.

For catadioptric systems that do not have a single effective viewpoint, or non-central cameras, the generation of geometrically correct perspective images is not possible. To illustrate this, consider the sketch in figure 2.3. The figure displays the combination of a conical mirror mounted in front of a perspective camera. The rays that are reflected by the mirror do not intersect at a single point in space if they had not been reflected. Suppose we wish to re-project the points $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$, which are the respective projections of scene points $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$ onto a sphere centered at an effective viewpoint. Let us require that the result of the re-projection is equivalent to a central projection onto a sphere from a desired effective viewpoint. For the sake of argument, let us choose $\mathbf{v}_{23}$ as the desired effective viewpoint. The dotted circle centered at $\mathbf{v}_{23}$ in the figure represents

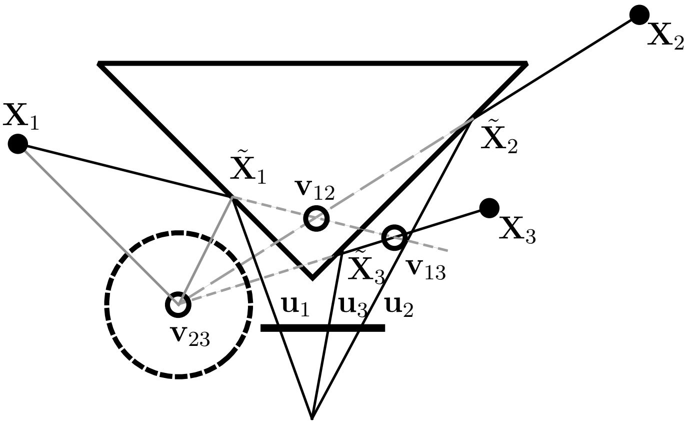


Figure 2.3: A conical mirror in front of a conventional camera does not yield a single effective viewpoint. The reflection rays from $\mathbf{u}_2$ and $\mathbf{u}_3$ intersect at $\mathbf{v}_{23}$. Point $\mathbf{v}_{23}$ is however not a single effective viewpoint of the catadioptric system; in general, the effective viewpoint for another pair of rays intersect at a different point. For example, the reflection rays for $\mathbf{u}_1$ and $\mathbf{u}_2$ intersect at $\mathbf{v}_{12}$. Hence, it is not possible to re-map the catadioptric image to a perspectively correct image formed at, say, the sphere centered at viewpoint $\mathbf{v}_{23}$. A correct re-projection of $\mathbf{u}_1$ is only possible when the distance to $\mathbf{X}_1$ is known.

the sphere onto which we wish to re-project. The mapping from points on the surface of the mirror to pixels is an invertible mapping. A pixel coordinate thus uniquely defines a point on the mirror surface and vice versa. Given $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$, their respective mirror surface points $\tilde{\mathbf{X}}_1$, $\tilde{\mathbf{X}}_2$ and $\tilde{\mathbf{X}}_3$ can be derived. Because $\mathbf{X}_2$, $\tilde{\mathbf{X}}_2$ and $\mathbf{v}_{23}$ are co-linear, the central projection of $\mathbf{X}_2$ onto the sphere is the same as that of $\tilde{\mathbf{X}}_2$. Similarly, the geometrically correct projection of $\mathbf{X}_3$ onto the sphere is uniquely determined by $\tilde{\mathbf{X}}_3$. Points $\mathbf{X}_1$, $\tilde{\mathbf{X}}_1$ and $\mathbf{v}_{23}$ are, however, not co-linear. In order to correctly re-project $\mathbf{u}_1$, the position of $\mathbf{X}_1$ along the ray passing through $\mathbf{X}_1$ and $\tilde{\mathbf{X}}_1$ has to be known. Because from $\mathbf{u}_1$ only $\tilde{\mathbf{X}}_1$ can be derived we can only guess where point $\mathbf{X}_1$ would project.

In [63] the family of mirror shapes whose members can — theoretically — all be used to construct central catadioptric vision systems is derived. If $z(r)$ is the profile of the mirror shape, where z denotes the height and $r = \sqrt{x^2 + y^2}$ is the radius, the complete family of mirror shaped is given by

$$\left(z - \frac{c}{2}\right)^2 - r^2 \left(\frac{t}{2} - 1\right) = \frac{c^2}{4} \left(\frac{t-2}{t}\right) \qquad (t \geq 2) \tag{2.1}$$

$$\left(z - \frac{c}{2}\right)^2 + r^2 \left(1 + \frac{c^2}{2t}\right) = \left(\frac{2t + c^2}{4}\right) \qquad (0 < t < 2), \tag{2.2}$$

where c denotes the distance between the pinhole of the camera and the effective viewpoint, and t is a constant of integration. These equations reveal that the mirror profiles form a 2-parameter (c and t) family of conic sections. Particular choices of the parameters yield different types of mirror shapes. However, some choices describe mirror shapes that cannot be used in practice to construct a central catadioptric panoramic camera.

For $t > 2$ and $c > 0$ a hyperboloid is obtained. A hyperboloid is defined by the locus of points for which the distance between two fixed points, called the foci F and F', is constant. When a hyperboloid is used to construct a catadioptric vision system, focus F lies inside the mirror. Rays that would have passed through F are reflected in such a way that they pass through the other focus F'. When it is ensured, by careful alignment of the camera and the mirror, that the pinhole R of the camera coincides with focal point F', an omnidirectional vision sensor with a single effective viewpoint at F is obtained. Figure 2.2 displays the hyperboloid mirror and its properties. Rees [76] was the first to realize a central catadioptric vision system based on a standard perspective camera and a hyperboloid mirror.

When $t \to \infty$, $c \to \infty$ and $c/t = h$ is constant, equation 2.1 describes a paraboloid. Rays passing through the focus of the paraboloid are reflected in a direction parallel to the mirror axis of symmetry. The paraboloid can be used to construct a central catadioptric system if the projection of the mirror into the image can be modeled by an orthographic, instead of a perspective, projection. This can be achieved using a telecentric lens, which can be regarded as a lens whose focal point lies at infinity. The paraboloid based system has several advantages over a hyperboloid based system. First, because the projection is orthogonal, the distance between the mirror and the lens is allowed to vary. Secondly, no internal reflections are caused by a transparent cylinder supporting the mirror because the reflected rays are all parallel to the cylinder axis of symmetry [38]. Finally, the system is easier to calibrate [27, 16, 46]. The major disadvantage is that telecentric lenses are expensive and relatively large.

Other solutions of equation 2.1 describe a plane, a sphere, a cone and an ellipse. If $t = 2$ and $c > 0$, equation 2.1 reduces to the equation of a plane. Planar mirrors do not increase the field of view; they only change the effective viewpoint. If $c = 0$ and $t > 0$, equation 2.1 describes a spherical mirror, and for $c = 0$ and $t \geq 2$ it describes a conical mirror. These mirror shapes cannot be used to construct a catadioptric system with a single effective viewpoint in practice because when $c = 0$ the effective pinhole and the effective viewpoint coincide. For the cone, this means that the pinhole is placed at the apex of the cone. The only rays that enter the effective pinhole from the mirror are the ones that graze the cone. In the spherical case, the effective viewpoint lies at the center of the sphere. The observer would therefore only observe itself and nothing else. Although the cone and the sphere cannot be used to construct a central catadioptric system, these mirror shapes have been employed in a wide range of mobile robot applications that do not critically depend on the single effective viewpoint property. Conic mirrors have been used in [111, 112, 113, 4]. Spherical mirrors have been used in [35, 108]. If $t > 0$ and $c > 0$, a concave ellipsoidal mirror is described. The maximum field of view covered by such a system is only half hemisphere because of self-occlusion. This is probably the reason that ellipsoid mirror based systems are hardly encountered in literature.

2.2 Geometric Image Formation

Our robot is equipped with a central catadioptric vision sensor consisting of a conventional camera and a hyperboloid mirror. In this section we describe its geometric image formation, which can be expressed as a sequence of coordinate transformations and central projections.

2.2.1 Notation and definitions

Points in 3-D space are represented by upper case letters, such as X. An upper case bold symbol, such as $\mathbf{X}$ refers to a Cartesian or homogeneous coordinate vector of X. The Cartesian vector is of the form $[X_1, X_2, X_3]^T$. The homogeneous vector is of the form $[X_4X_1, X_4X_2, X_4X_3, X_4]^T$, where X_4 is an arbitrary non-zero scalar. When $X_4 = 1$, the homogeneous vector is said to be normalized. A normalized homogeneous vector is obtained by dividing a homogeneous vector by its 4th component. For convenience, we define the operator $\mathcal{N}$ that performs the division, i.e. $[X_1, X_2, X_3, 1]^T = \mathcal{N}[X_4X_1, X_4X_2, X_4X_3, X_4]^T$. Several Cartesian coordinate frames will be defined to describe the geometric image formation. A coordinate frame (position and orientation) whose origin coincides with a point Q will be referred to as Q. Vectors with a subscript, such as $\mathbf{X}_Q$, represent vectors measured in coordinate frame Q. A pure translation is represented as 4×4 matrix $\mathbf{T}$ of the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{2.3}$$

where $\mathbf{I}_3$ denotes the 3×3 identity matrix, $\mathbf{0}$ denotes the 3×1 null vector, and $\mathbf{t}$ denotes a 3×1 translation vector. A pure rotation is represented as a 4×4 matrix $\mathbf{R}$ of the form

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_3 & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{2.4}$$

where $\mathbf{R}_3$ is a 3×3 rotation matrix. A rigid transformation is represented by a 4×4 matrix $\mathbf{M} = \mathbf{TR}$. Subscripts, such as in $\mathbf{M}_{QR}$, are used to denote a change in coordinate frames $\mathbf{X}_Q = \mathbf{M}_{QR}\mathbf{X}_R$.

The upper case letter X is reserved to denote scene points. An upper case bold $\mathbf{X}$ refers to its coordinates. We use $\tilde{\mathbf{X}}$ to refer to the coordinates of the projection of X onto the mirror. A lower case $\mathbf{x}$ refers to the projection of X (via the mirror) to the normalized image plane of the perspective camera. Finally, a lower case $\mathbf{u}$ refers to pixel coordinates of a projected scene point X.

We define two important coordinate frames: the mirror coordinate frame M and the camera coordinate frame R. The mirror coordinate frame is centered at F, the focal point inside the mirror, and we assume its Z-axis coincides with the mirror axis of symmetry. The camera coordinate frame R is centered at F', the other focus of the hyperboloid. The coordinate frames and their relationships are illustrated in figure 2.4.

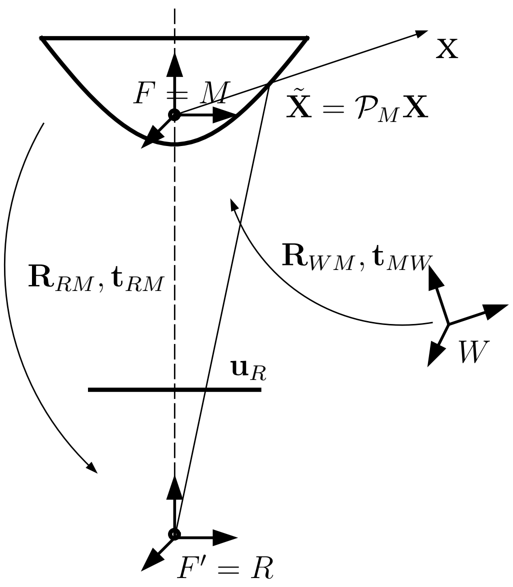

X
$F = M$
$\tilde{\mathbf{X}} = \mathcal{P}_M\mathbf{X}$
$\mathbf{R}_{RM}, \mathbf{t}_{RM}$
$\mathbf{R}_{WM}, \mathbf{t}_{MW}$
W
$\mathbf{u}_R$
$F' = R$

Figure 2.4: Coordinate frames and their relationships. The mirror coordinate frame is centered at F, the focal point in the mirror. The pinhole of the perspective camera coincides with the origin of the camera coordinate frame, which is centered at the other focal point F'.

2.2.2 Hyperboloid image formation

A hyperboloid mirror is defined in the mirror coordinate frame M centered at the mirror focus F by the equation

$$\frac{(z+e)^2}{a^2} - \frac{x^2+y^2}{b^2} = 1, \tag{2.5}$$

where a and b are parameters whose ratio governs the shape of the mirror, and $e = \sqrt{a^2+b^2}$ is the eccentricity of the mirror. The focal points F and F' are separated by a distance $2e$. See figure 2.5.

The central projection of point X onto point $\tilde{X}$ on the surface of the mirror is as follows. Let $\mathbf{X}_M = [X_1, X_2, X_3]^T$ denote the coordinate of point X specified in mirror coordinate frame M. The ray from F to X can be specified by $\lambda\mathbf{X}_M$. We seek λ such that $\lambda\mathbf{X}_M = \tilde{\mathbf{X}}_M$. Substituting the ray equation $\lambda\mathbf{X}_M$ into equation 2.5 gives

$$\frac{(\lambda X_3+e)^2}{a^2} - \frac{(\lambda X_1)^2+(\lambda X_2)^2}{b^2} = 1. \tag{2.6}$$

Expressed in quadratic form, the above reads

$$\lambda^2(b^2X_3^2 - a^2X_1^2 - a^2X_2^2) + \lambda(b^2 2eX_3) + b^4 = 0. \tag{2.7}$$

Solving for λ results in two solutions for λ,

$$\lambda_1, \lambda_2 = \frac{b^2(-eX_3 \pm a\,\|\mathbf{X}\|)}{b^2X_3^2 - a^2(X_1^2+X_2^2)}. \tag{2.8}$$

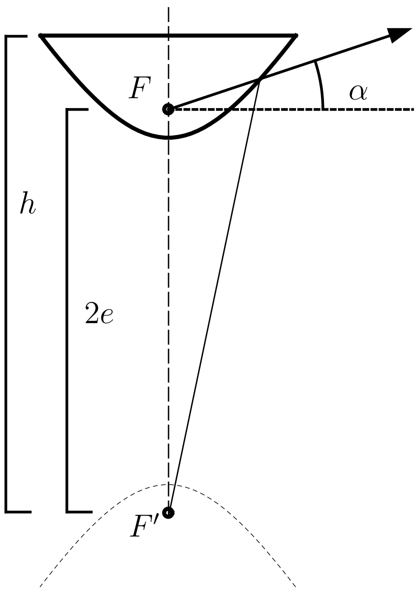


Figure 2.5: The hyperboloid and its parameters. In order to obtain a single effective viewpoint, the distance between the focal point F inside the mirror and the focal point F' of the camera should equal a constant value $2e$ (which follows from the mirror shape). The finite height h limits the maximum vertical viewing angle α that can be obtained.

Figure 2.6 graphically displays three possible combinations of the signs of λ_1 and λ_2. If a point $\mathbf{X}$ is located in the area marked by ++, the line passing through F and $\mathbf{X}$ intersects both the actual mirror, and a "virtual mirror", which is centered at F'. This "virtual mirror" corresponds to the other sheet of the hyperboloid defined by equation 2.5. The intersection closest to F is the intersection with the actual mirror, i.e. we pick $\lambda = \min(\lambda_1, \lambda_2)$. When the signs of λ_1 and λ_2 are different, the positive solution describes the intersection between F and X. The negative solution describes the intersection with the other sheet. In this case, we pick $\lambda = \max(\lambda_1, \lambda_2)$. Finally, when both λ_1 and λ_2 are negative, the point lies inside the mirror and thus cannot be observed. The selection of the correct value of λ can be expressed more formally as

$$\lambda = \begin{cases} \min(\lambda_1, \lambda_2) & \text{if } \lambda_1 > 0 \text{ and } \lambda_2 > 0 \\ \max(\lambda_1, \lambda_2) & \text{if } \operatorname{sign}(\lambda_1) \neq \operatorname{sign}(\lambda_2) \\ \text{no intersection} & \text{otherwise} \end{cases} . \tag{2.9}$$

We define an operator $\mathcal{P}_M$ that performs the central projection of a point onto the hyperboloid as

$$\mathcal{P}_M \mathbf{X}_M = \begin{bmatrix} X_4 & 0 & 0 & 0 \\ 0 & X_4 & 0 & 0 \\ 0 & 0 & X_4 & 0 \\ 0 & 0 & 0 & r \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} , \tag{2.10}$$

where $r = 1/\lambda$, and λ is obtained from the vector $[X_1, X_2, X_3]$ using equations 2.8 and 2.9. Given a point X, we can derive its central projection $\tilde{X}$ onto the mirror surface.

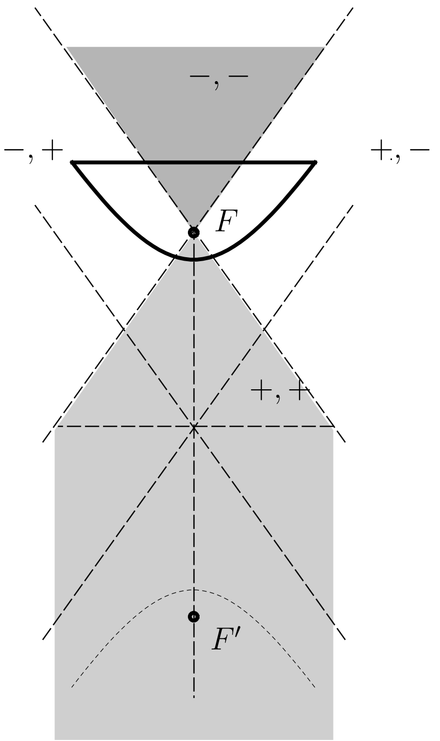


Figure 2.6 : Intersections of a ray with the mirror.

A second central projection describes the mapping from points on the mirror surface into the image. The mapping from mirror coordinates $\tilde{\mathbf{X}}_M = \mathcal{P}_M \mathbf{X}_M$ to homogeneous pixel coordinates $\mathbf{u}_R$ in the image involves several coordinate transformations and a projection. First, points expressed in the mirror coordinate frame are rotated and translated into the camera coordinate frame. Then, the points are projected onto the normalized image plane of the pinhole camera. Finally, an affine mapping brings the normalized image coordinates into pixel coordinates. This sequence can be expressed as

$$\mathbf{u}_C = \mathbf{K}_R \mathcal{P}_P \mathbf{M}_{RM} \mathbf{X}_M, \tag{2.11}$$

where $\mathbf{K}_R$ is a 3×3 upper triangular camera calibration matrix that maps coordinates on the normalized image plane to pixel coordinates, $\mathcal{P}_P = [\mathbf{I}_3 \quad \mathbf{0}]$ is the perspective projection matrix that projects points onto the plane $Z = 1$, and $\mathbf{M}_{RM}$ is the rigid transformation (rotation and translation) relating the mirror frame M and the camera frame C. For details on the camera model, the reader is referred to [17]. The single effective viewpoint property of the catadioptric system is preserved under rotation of the camera or the mirror about their respective focal points because these keep the distance separating the camera pinhole and the effective viewpoint constant. It is however a common practice to align the mirror axis of symmetry with the camera Z-axis. In this way, the maximal elevation viewing angle (with respect to the camera frame) is the same for all points on the mirror edge.

We have shown how points expressed in the mirror coordinate frame can be transformed

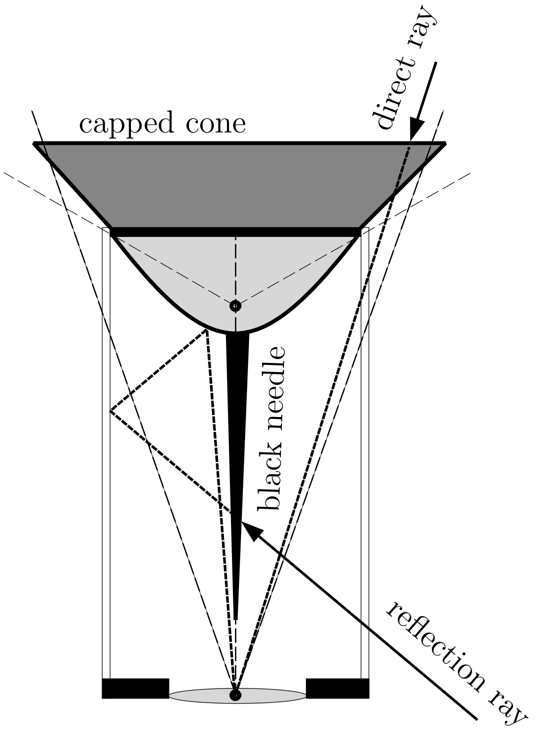


Figure 2.7: A black needle coinciding with the axis of symmetry of the glass cylinder supporting the mirror prevents internal reflections. A capped cone on top of the mirror obstructs the view of anything above the camera that is outside the field of view covered by the mirror. The capped cone reduces the effects of automatic gain correction.

to points expressed in the pixel coordinate frame. Introducing a global world coordinate frame W, the projection of a point X represented by $\mathbf{X}_W$ to point u represented by the normalized homogeneous vector $\mathbf{u}_R$ in the omni-directional image can concisely be written as

$$\mathbf{u}_C = \mathbf{K}_R \mathcal{P}_P \mathbf{M}_{RM} \mathcal{P}_M \mathbf{M}_{MW} \mathbf{X}_W. \tag{2.12}$$

We have seen that the image formation of our catadioptric vision sensor can be described as a composition of two central projections and a number of coordinate transformations. The first central projection maps a point in the world to a point on the surface of the mirror. The second central projection maps a point on the surface of the mirror to a point in the image.

2.3 Design and Calibration

2.3.1 Our mirror

Our hyperbolic mirror is manufactured by Accowle [1]. The mirror is made from aluminium and is supported by a glass hollow cylinder. A black needle extends from the mirror. The needle coincides with the main axis of the glass cylinder. Light that reflects on the internal surface of the cylinder and then passes through the camera center of projection crosses the main axis of the cylinder. The needle eliminates such internal reflections. A diagram of the sensor is shown in figure 2.7.

Because our camera performs automatic gain correction to prevent clipping of intensities, drastic changes in the overall image intensity occur when lights mounted in the ceiling enter or leave the field of view. As a simple but effective solution, we placed a capped cone on top of the omnidirectional vision system. The cone remains just outside the field of view of the mirror and is large enough to occupy the entire field of view of the perspective camera. As a result, light from above is blocked by the cone.

2.3.2 Design of the mirror

Several considerations guide the design of a practical mirror; its dimensions and weight, its coverage of the view sphere and the height at which the mirror has to be mounted with respect to the focal point of the camera.

In this section we explain how a mirror can be designed that meets a set of user specified requirements. A first requirement is that the projection of the mirror should occupy the whole image. In our derivation, we take the view that the mirror is to be used in conjunction with a camera, which has a known focal length. Given a user specified mirror radius, this assumption results in a constraint on the height of the mirror. A user specified maximal elevation angle then suffices to derive the mirror parameters a and b. The reader should note that a similar derivation can be performed when taking the view that the height and the mirror radius are specified by the user. These choices then result in a constraint on the focal length of the lens.

To simplify the equations in our derivation, we suppose that the camera has no skew and unit aspect ratio. Under this assumption, the mirror edge projects to a circle in the image with pixel radius r_{pix}. The desired pixel radius is chosen close the half the number of rows in the image in order to achieve the maximal resolution. Given a choice of the mirror rim radius r, the height of the mirror rim above the effective pinhole can be derived using similarity of triangles as

$$h = f\frac{r}{r_{\text{pix}}}, \tag{2.13}$$

where f is the known focal length in pixels.

The next step is to choose a desired maximal elevation angle α (see figure 2.5 for an illustration). The maximal elevation angle is attained for rays intersecting the mirror rim. The height of the mirror rim, expressed with respect to the focal point F can then be derived as

$$z = r\tan\alpha. \tag{2.14}$$

Because the distance separating the two focal points of the hyperboloid is $2e$, the value of e can be derived from the known height of the mirror rim h and z as

$$e = \frac{h - z}{2}. \tag{2.15}$$

The mirror parameters a and b can now be derived. The parameters a and b are related by a single parameter t, which can be derived from equation 2.1 as

$$a = e\sqrt{\frac{t-2}{t}}, \quad b = e\sqrt{\frac{2}{t}}, \tag{2.16}$$

where we have substituted $e = c/2$. If we substitute $s = 2/t$ in the previous equation, and substitute the coordinates $(z, r) = (h - 2e, r)$ of a point on the mirror edge in the mirror equation, the mirror equation reads

$$\frac{(h-e)^2}{e^2(1-s)} - \frac{r^2}{e^2 s} = 1, \tag{2.17}$$

which can be expressed in quadratic form as

$$s^2e^2 + s(h^2 - 2eh + r^2) - r^2 = 0. \tag{2.18}$$

The solutions for s are given by

$$s = \frac{-(h^2 - 2eh + r^2) \pm \sqrt{(h^2 - 2eh + r^2)^2 + 4e^2r^2}}{2e^2}. \tag{2.19}$$

Finally, the parameters a and b are determined by plugging in the value $t = 2/s$ in equation 2.16.

2.3.3 Calibration of the vision sensor

Camera calibration is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and the 3-dimensional position and orientation of the camera frame relative to a certain frame of reference (extrinsic parameters). Calibration of the omni-directional vision sensor can be done in two steps. First, the perspective camera can be calibrated by a conventional camera calibration method. Next, the camera is positioned with respect to the mirror in such a way that the single effective viewpoint property is obtained.

To calibrate the camera, 3-dimensional coordinates of reference control points on a calibration target and corresponding 2-D coordinates of the image observation are required. We use a checker-board patterned calibration target. The corners of the squares act as control points. They can be detected accurately with little user interaction. Camera calibration involves minimizing the error between measured positions of control points and the positions of the control points as predicted by the camera model as a function of the camera model parameters. Various calibration methods have been presented in literature. Linear camera calibration methods assume a linear camera model (i.e. the pinhole model). While calibration is fast (no iterations are required) the accuracy is often poor because the model is too simplistic. The best known linear camera calibration method is due to Tsai [97, 98, 107]. More realistic camera models include non-linear terms accounting for lens distortion. We adopt the camera model presented in [33], which includes lens distortion. This model requires a non-linear optimization. The optimization routine may get trapped in a local minimum of the error function. To decrease the probability of getting trapped in a local minima, a good initial estimate of the model parameters is needed. The risk of getting trapped is reduced by first using a linear technique to provide an initial estimate of the model parameters. This estimate is subsequently refined using non-linear optimization of all parameters.

The next step involves positioning the camera with respect to the mirror in such a way that the single effective viewpoint property is obtained. Svoboda [84] proposes the following method. It is assumed that the mirror parameters are accurately known from manufacturing, and that the intrinsic camera parameters have been estimated reliably. By design, the desired height and the radius of the mirror rim are known. The perspective camera model can then be used to predict the image of the mirror edge as it should be observed when the camera is positioned correctly with respect to the mirror. By overlaying the prediction in a live video window, the camera position can be adjusted manually so as to obtain an accurate registration of the observed and the predicted mirror rim. We employed this strategy when we worked with a prototype of an omnidirectional vision sensor we designed and manufactured ourselves.

The support for the mirror produced by Accowle [1] is screwed directly on the C-mount of the CCD camera. This leaves no degrees of freedom in positioning the mirror with respect to the camera. However, our camera does not have a fixed focus lens but is capable of zooming. The zoom setting (focal length) of the camera is motor controlled. It is initialized to a default setting on camera power up. The focal length of the camera may therefore vary slightly from session to session. Furthermore, the principal point (projection of the camera frame Z-axis into the image) is known to be difficult to estimate reliably. Performing a full fledged camera calibration each session is a time consuming and cumbersome process. Instead, we adopt the following simple calibration scheme that re-estimates only the focal length and the principal point of the camera.

A full camera calibration returns (amongst other parameters) a camera calibration matrix of the form

$$\mathbf{K} = \begin{bmatrix} f_u & \gamma f_u & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.20}$$

where f_u and f_v encode the focal lengths (a unique value in meters) expressed in units of horizontal and vertical pixels, γ encodes skew between the sensor axes, and (c_u, c_v) encodes the principal point location. If we let $\beta = f_v/f_u$ denote the aspect ratio, then the calibration matrix may be decomposed as

$$\mathbf{K} = \mathbf{SA} = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_u & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.21}$$

where $\mathbf{S}$ is a similarity transformation, and $\mathbf{A}$ is an affine transformation. We re-estimate the coefficients in $\mathbf{S}$, while fixing the ones contained in $\mathbf{A}$ to their estimates obtained from the full camera calibration. In particular, a set of K points $\mathbf{u}_k$ lying on the mirror rim are manually selected in an omnidirectional image using the mouse pointer. The inverse of the affine transformation $\mathbf{A}$ is applied to undo the skew and non-uniform scaling. This gives a new set of points $\mathbf{u}'_k = \mathbf{A}^{-1}\mathbf{u}_k$. A circle of the form $(u' - c_u)^2 + (v' - c_v) = r^2$, where (c_u, c_v) denotes the center of the circle and r its radius, can then be fitted to the points. The center of the circle is used as an estimate of the principal point. Its radius can be used to derive an estimate of the focal length of the camera.

The parameters of the circle can be estimated linearly from the points $\mathbf{u}'_k$ by solving the following linear system of equations

$$\begin{bmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots \\ u_k & v_k & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -u_1^2 - v_1^2 & & \\ -u_2^2 - v_2^2 & & \\ \vdots & \vdots & \vdots \\ -u_k^2 - v_k^2 & & \end{bmatrix}, \tag{2.22}$$

where $a = -2c_u$, $b = -2c_v$ and $c = c_u^2 + c_v^2 - r^2$. The center of the fitted circle, corresponding to the re-estimated principal point location, is derived from the estimated parameters as $(c_u, c_v) = (-a/2, -b/2)$. Its radius can be calculated as $r = \sqrt{c_u^2 + c_v^2 - c}$. Using similarity of triangles, the focal length f_u can be re-calculated as $f_u = rh_{rim}/r_{rim}$, where r is the circle measured circle radius, h_{rim} is the height of the mirror rim and r_{rim} is the radius of the mirror rim.

2.4 Virtual Cameras

In this section we introduce several virtual cameras. The virtual cameras are obtained by re-projecting the catadioptric image onto different surfaces. A virtual spherical camera, a virtual cylindrical camera and a virtual planar camera are developed. The spherical camera implements a re-projection of the catadioptric image onto a unit sphere centered at the effective pinhole. The cylindrical camera re-projects onto a cylinder. The planar camera re-projects onto a plane. The primary reason for developing these virtual cameras is that certain tasks become easier. For example, lines in the world project to lines in a planar perspective image, to sinusoids in a cylindrical panoramic image, whereas they project to general conics in the catadioptric image.

Each virtual camera has its own coordinate frame; C denotes the coordinate frame associated with a virtual cylindrical camera, P is used for a virtual planar perspective camera and S is associated with a spherical camera. All virtual cameras share the same focal point, which coincides with F. However, they may differ in orientation.

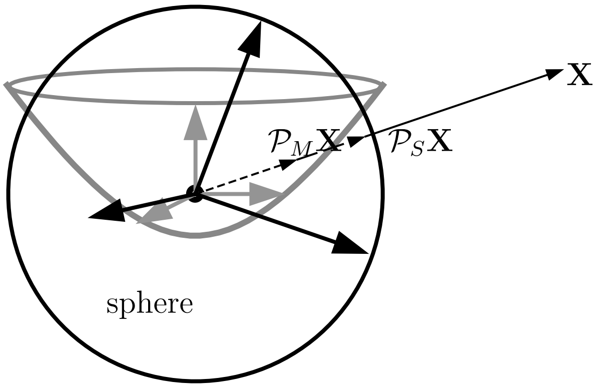


Figure 2.8: A virtual spherical panoramic camera is defined by a sphere centered at the effective viewpoint F. The mirror coordinate frame and the spherical panoramic camera coordinate frame are related by a rotation.

2.4.1 Virtual spherical camera

A spherical perspective image is obtained by projecting the environment onto a sphere and unfolding the sphere to a rectangular grid of pixels. The equation of a unit sphere centered at the origin is given by

$$x^2 + y^2 + z^2 = 1. \tag{2.23}$$

The central projection of X onto a point X' on the surface of the unit sphere can be described as follows. Let $\mathbf{X}_S = [X_1, X_2, X_3]^T$ denote the coordinate of point X specified in coordinate system S. The ray from F through X is specified by $\lambda\mathbf{X}$. We seek λ such that $\lambda\mathbf{X} = \tilde{\mathbf{X}}$, where $\tilde{\mathbf{X}}$ is a point on the surface of the sphere. The λ for which the ray intersects the sphere is found by solving the quadratic equation

$$\lambda^2(X_1^2 + X_2^2 + X_3^2) - 1 = 0, \tag{2.24}$$

which is obtained by substituting $\lambda\mathbf{X}$ into 2.23. Solving for λ gives two solutions:

$$\lambda_{1,2} = \pm\frac{1}{\sqrt{X_1^2 + X_2^2 + X_3^2}}. \tag{2.25}$$

The solution describing the point of intersection between F and X is given by

$$\lambda = \max(\lambda_1, \lambda_2). \tag{2.26}$$

We define an operator $\mathcal{P}_S$ such that $\tilde{\mathbf{X}} = \mathcal{P}_S\mathbf{X}$ that performs the central projection of a point onto the sphere,

$$\mathcal{P}_S\mathbf{X} = \begin{bmatrix} X_4 & 0 & 0 & 0 \\ 0 & X_4 & 0 & 0 \\ 0 & 0 & X_4 & 0 \\ 0 & 0 & 0 & r \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}, \tag{2.27}$$

where $r = 1/\lambda$ is obtained from equations 2.25 and 2.26.

The coordinates of $\tilde{X}$ can be expressed in a spherical coordinate representation $\mathbf{x} = [\varphi, \theta, 1]$, where φ denotes the angular elevation, θ denotes the angular azimuth, and 1 denotes the unit radius. We define an operator $\mathcal{S}$ that transforms a normalized homogeneous vector $\mathbf{X}$ representing a point on the unit sphere to $\mathbf{x}$,

$$\mathcal{S}\mathbf{X} = \begin{bmatrix} \operatorname{arctan2}(X_2, X_1) \\ \operatorname{arctan2}(X_3, \sqrt{X_1^2 + X_2^2 + X_3^2}) \\ 1 \end{bmatrix}. \tag{2.28}$$

The inverse $\mathcal{S}^{-1}$ is given by

$$\mathcal{P}_S^{-1}[\varphi, \theta, 1]^T = \begin{bmatrix} \cos\theta\cos\varphi \\ \cos\theta\sin\varphi \\ \sin\theta \\ 1 \end{bmatrix}. \tag{2.29}$$

Normalized homogeneous spherical image pixel coordinates $\mathbf{u}_S$ are related to $\mathbf{X}_S$ by

$$\mathbf{u}_S = \mathbf{K}_S \mathcal{S} \mathcal{N} \mathcal{P}_S \mathbf{X}_S, \tag{2.30}$$

where $\mathbf{K}_S$ is a 3×3 upper triangular calibration matrix, which maps spherical coordinates to pixel coordinates.

The calibration matrix implements a scaling and a translation of coordinates and can be derived as follows. Let a be in an interval $[a_{\min}, a_{\max}]$. Let b be in an interval $[b_{\min}, b_{\max}]$. Let $\Delta a = a_{\max} - a_{\min}$ and $\Delta b = b_{\max} - b_{\min}$. The conversion of a coordinate v_a expressed in a units to a coordinate v_b expressed in b units can be written as

$$\begin{aligned} v_b &= (v_a - a_{\min})\frac{\Delta b}{\Delta a} + b_{\min} \\ &= v_a \frac{\Delta b}{\Delta a} + b_{\min} - a_{\min}\frac{\Delta b}{\Delta a}. \end{aligned} \tag{2.31}$$

We now derive the calibration matrix for a spherical camera whose field of view covers the azimuth range $\phi \in [-\pi, \pi]$, and whose elevation is in the range $\theta \in [-\alpha, \alpha]$, where α denotes the maximal elevation angle of the hyperbolic mirror. The rows in a spherical image correspond to the elevation θ. The columns correspond to the azimuth φ. Let N_c and N_r denote the number of columns and rows in the spherical image. Given the specified number columns, we calculate the number of rows required to get approximately square pixels from the following ratio:

$$\frac{\alpha\pi}{\pi} = \frac{N_r}{N_c}. \tag{2.32}$$

Using these ratios, the number of rows N_r is calculated as

$$N_r = \operatorname{round}(N_c \alpha). \tag{2.33}$$

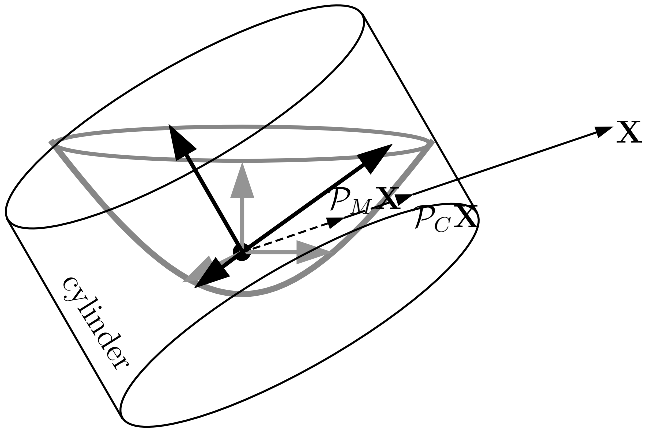


Figure 2.9: A virtual cylindrical panoramic camera is defined by a cylinder centered at the effective viewpoint F. The mirror coordinate frame and the cylindrical panoramic camera coordinate frame are related by a rotation.

The camera calibration can then be derived as

$$\begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.34}$$

where $f_u = N_r/2\pi$, $f_v = N_c/2\alpha$, $u_c = \pi N_r/2\pi$ and $u_v = \alpha N_c/2\alpha$, which can be derived using equation 2.31.

The mapping from pixel coordinates to points on the sphere is given by

$$\mathbf{x}_S = \mathcal{S}^{-1}\mathbf{K}_S^{-1}\mathbf{u}_S. \tag{2.35}$$

Generating an image obtained by a virtual spherical camera involves mapping pixel coordinates $\mathbf{u}_S$ in the spherical image to pixel coordinates in $\mathbf{u}_R$ the image acquired by the real camera. The mapping can be expressed as

$$\mathbf{u}_R = \mathbf{K}_R\mathcal{P}_P\mathbf{M}_{RM}\mathcal{P}_M\mathbf{R}_{MS}\mathcal{S}^{-1}\mathbf{K}_S^{-1}\mathbf{u}_S. \tag{2.36}$$

2.4.2 Virtual cylindrical camera

A cylindrical image can be obtained by projecting the environment onto a cylinder and unfolding the cylinder to a rectangular grid of pixels.

The equation of a cylinder with unit radius whose axis of symmetry coincides with the z-axis is given by

$$x^2 + y^2 = 1. \tag{2.37}$$

The central projection of X onto a point X' on the surface of the cylinder can be described as follows. Let $\mathbf{X}_C = [X_1, X_2, X_3]^T$ denote the coordinate of point X specified in coordinate system C. The ray from F through X is specified by $\lambda\mathbf{X}$. We seek λ such that $\lambda\mathbf{X} = \tilde{\mathbf{X}}$, where $\tilde{\mathbf{X}}$ denotes a point on the surface of the cylinder. The λ for which the ray intersects the sphere is found by solving the quadratic equation

$$\lambda^2(X_1^2 + X_2^2) - 1 = 0, \tag{2.38}$$

which is obtained by substituting $\lambda\mathbf{X}$ into 2.37. Solving for λ gives two solutions:

$$\lambda_{1,2} = \pm\frac{1}{\sqrt{X_1^2 + X_2^2}}. \tag{2.39}$$

The solution that describes the point of intersection between F and X is given by

$$\lambda = \max(\lambda_1, \lambda_2). \tag{2.40}$$

We define an operator $\mathcal{P}_C$ that performs the central projection of a point onto the cylinder,

$$\mathcal{P}_C\mathbf{X} = \begin{bmatrix} X_4 & 0 & 0 & 0 \\ 0 & X_4 & 0 & 0 \\ 0 & 0 & X_4 & 0 \\ 0 & 0 & 0 & r \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}, \tag{2.41}$$

where $r = 1/\lambda$ is obtained from equations 2.39 and 2.40.

The coordinates of $\tilde{X}$ can be expressed in a cylindrical coordinate representation $\mathbf{x} = [z, \theta, 1]$, where z denotes the elevation, θ denotes the angular azimuth, and 1 denotes the unit radius. We define an operator $\mathcal{C}$ that transforms a normalized homogeneous vector $\tilde{\mathbf{X}}$ representing a point on the unit sphere to $\mathbf{x}$,

$$\mathcal{C}\mathbf{X} = \begin{bmatrix} \arctan2(X_2, X_1) \\ \frac{X_3}{\sqrt{X_1^2+X_2^2}} \\ 1 \end{bmatrix}. \tag{2.42}$$

The inverse $\mathcal{C}^{-1}$ is given by

$$\mathcal{C}^{-1}[\varphi, \theta, 1]^T = \begin{bmatrix} \cos\theta \\ \sin\theta \\ z \\ 1 \end{bmatrix}. \tag{2.43}$$

Normalized homogeneous cylindrical pixel coordinates $\mathbf{u}_C$ can be related to $\mathbf{X}_C$ via

$$\mathbf{u}_C = \mathbf{K}_C\mathcal{C}\mathcal{N}\mathcal{P}_C\mathbf{X}_C. \tag{2.44}$$

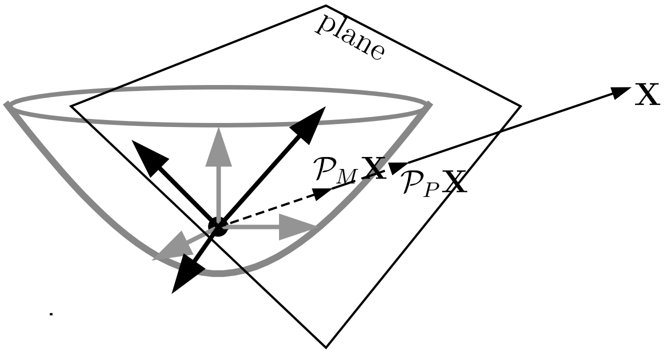


Figure 2.10: A virtual planar perspective camera is defined by a plane at unit distance from the effective viewpoint F. The mirror coordinate frame and the virtual planar perspective camera coordinate frame are related by a rotation.

where K is a 3×3 upper triangular calibration matrix that maps cylindrical coordinates to pixel coordinates. The form of the calibration matrix can be derived as explained in the previous paragraph.

The rows in a panoramic image correspond to the elevation z. The columns correspond to the azimuth φ. Let N_c and N_r denote the number of columns and the number of rows in a panoramic image. From the specified number of columns, we calculate the number of rows required to get approximately square pixels from the following ratio,

$$\frac{h-2e}{\pi r_{\text{rim}}} = \frac{N_r}{N_c}, \tag{2.45}$$

which is based on the ratio of the height of the mirror rim with respect to the focal point inside the mirror and half the edge of the mirror rim. Using the ratio, the number of rows N_r is calculated as

$$N_r = N_c \frac{\pi r_{\text{rim}}}{h-2e}. \tag{2.46}$$

The mapping from pixel coordinates to points on the cylinder is given by

$$\mathbf{x}_C = \mathcal{C}^{-1}\mathbf{K}_C^{-1}\mathbf{u}_C. \tag{2.47}$$

Generating an image obtained by a virtual cylindrical camera involves mapping cylindrical pixel coordinates $\mathbf{u}_S$ to pixel coordinates $\mathbf{u}_R$ in the image acquired by the real camera. The mapping can be expressed as

$$\mathbf{u}_R = \mathbf{K}_R\mathcal{P}_P\mathbf{M}_{RM}\mathcal{P}_M\mathbf{R}_{MC}\mathcal{C}^{-1}\mathbf{K}_C^{-1}\mathbf{u}_C. \tag{2.48}$$

2.4.3 Virtual planar perspective camera

Pixel coordinates $\mathbf{u}$ and coordinates on the normalized image plane $\mathbf{x}$ are related via

$$\mathbf{u}_P = \mathbf{K}_P\mathbf{x}_P. \tag{2.49}$$

A virtual perspective camera can be specified by its calibration matrix, $\mathbf{K}_P$. The calibration matrix for an $N_r \times N_c$ image obtained by a camera with zero skew and square pixels, covering a field of view of γ degrees, can be expressed as:

$$\mathbf{K} = \begin{bmatrix} N_c/2\cot(\gamma/2) & 0 & N_c/2 \\ 0 & N_c\cot(\gamma/2) & N_r/2 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.50}$$

The mapping from virtual planar perspective image pixel coordinates to pixel coordinates in the image acquired by the camera can be expressed as

$$\mathbf{u}_R = \mathbf{K}_R \mathcal{P}_P \mathbf{R}_{RP} \mathcal{P}_P^{-1} \mathbf{K}_P^{-1} \mathbf{u}_P, \tag{2.51}$$

where $\mathcal{P}_P^{-1}$ extends the vector with a homogeneous component whose value equals 1.

2.4.4 Re-sampling issues

In the previous paragraphs we have shown how the coordinates of pixels in a virtual image are mapped to pixel coordinates in a catadioptric image. The presented mappings specify the locations where the input image is to be re-sampled. Generating an output image can be done in a straightforward manner; each pixel in the output image is mapped to a point in the input image (catadioptric image). The intensity value at the mapped point is then assigned to the respective output pixel. There is one issue however; the re-sampling grid does not coincide with the input sampling grid, which is taken to be the integer lattice. What value should be assigned to input coordinates that lie in-between pixels? The solution is to convert the discrete input image into a continuous surface, a process known as reconstruction or interpolation. Once the input image is reconstructed it can be sampled at any position.

In our re-sampling implementation we use bi-linear interpolation to reconstruct the catadioptric input image. The bi-linear interpolation method is as follows. Suppose $\mathbf{u} = (u, v)$, where u and v are integer coordinates, map to $\mathbf{u}_R = (u_R, v_R)$ in the catadioptric image. Bi-linear interpolation uses the image intensities at the four pixels (u_{R0}, v_{R0}), (u_{R1}, v_{R0}) (u_{R0}, v_{R1}), (u_{R1}, v_{R1}) which are closest to (u_R, v_R) in the catadioptric image:

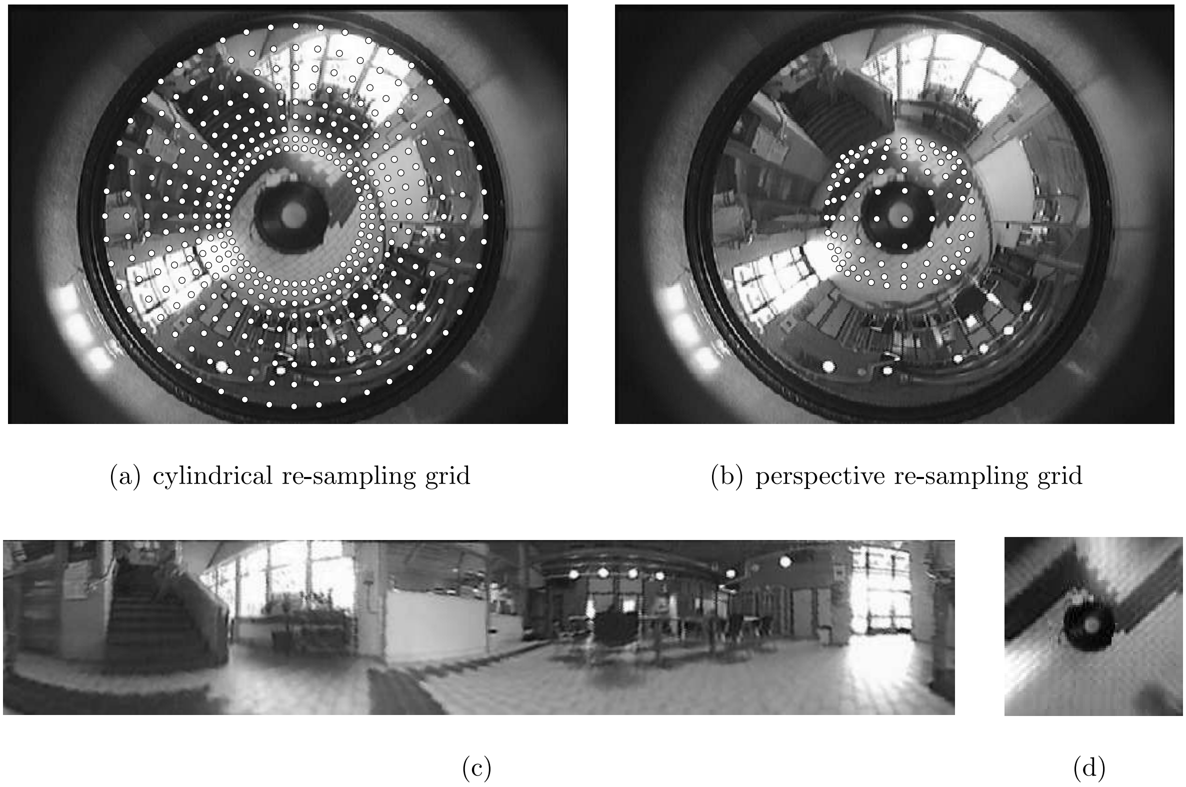
$$u_{R0} = \text{floor}(u_R), \quad u_{R1} = u_{R0} + 1, \quad v_{R0} = \text{floor}(v_R), \quad v_{R1} = v_{R0} + 1. \tag{2.52}$$

The intensity values are interpolated along the u-axis to produce two intermediate results I_0 and I_1,

$$I_0 = I_R(u_R, v_{R0}) = I_R(u_{R0}, v_{R0})(u_{R1} - u_R) + I_R(u_{R1}, v_{R0})(u_R - u_{R0}), \tag{2.53}$$
$$I_1 = I(u_R, v_{R1}) = I_R(u_{R0}, v_{R1})(u_{R1} - u_R) + I_R(u_{R1}, v_{R1})(u_R - u_{R0}), \tag{2.54}$$

where $I(u, v)$ denotes the intensity at pixel (u, v) in the catadioptric image. Then, the intensity $I(u_R, v_R)$ is computed by interpolating the intermediate values I_0 and I_1 along the v-axis:

$$I(u, v) = I_0(v_{R1} - v_R) + I_1(v_R - v_{R0}). \tag{2.55}$$


(a) cylindrical re-sampling grid

(b) perspective re-sampling grid

(c)

(d)

Figure 2.11: Course re-sampling grids and re-sampled images. a) re-sampling grid for forming a cylindrical panoramic image, b) re-sampling grid for forming a perspective image, c) cylindrical panoramic image, d) perspective image. The re-sampling grids illustrate that proper re-sampling requires smoothing with space-variant filtering kernels to prevent aliasing.

The advantage of bi-linear interpolation over other interpolation methods is that it is computationally inexpensive. A disadvantage of the method is that at places where the input image is under-sampled, aliasing occurs. Aliasing results in spurious resolution; the output image has high-frequency components that are not present in the input image.

To overcome this issue one can either increase the sampling rate, or bandlimit the input image [110]. The first solution is ideal but costly. The second solution forces the input image to conform to the low sampling rate by attenuating the high frequency components that give rise to aliasing artifacts. The suppression of high-frequency components can be done by filtering the input image with a low-pass filter such as a Gaussian kernel. Figure 2.11a displays a course re-sampling grid used to transform the catadioptric image into the cylindrical panoramic image shown in figure 2.11c. We see that the sampling rate decreases as we get further from the image center. Figure 2.11b displays a course re-sampling grid used to transform the catadioptric image into a perspective image shown in figure 2.11d. In this case, the re-sampling rate increases as we get further from the image center. Both cases illustrate that a proper re-sampling of catadioptric images into other perspective image representations requires smoothing of the input image with a kernel whose shape and size varies with position (space-variant filtering).

2.5 Conclusions

In this chapter we have discussed a central panoramic catadioptric vision sensor. We have reviewed various methods and sensor designs aimed toward acquiring panoramic imagery. Roughly, approaches can be categorized as image mosaicing (using images acquired by a single rotating camera), dedicated hardware (often employing multiple cameras arranged in a compact framework), and catadioptric vision sensors.

Catadioptric vision sensors employ a curved mirror that is mounted in front of a camera lens. Such sensors are particularly well suited for mobile robot applications because they instantly provide a panoramic image, are compact, relatively cheap, consume little power, and suffer little from wear-and-tear because they have no moving parts. Of particular interest are catadioptric systems that respect the single effective viewpoint constraint. The single effective viewpoint constraint expresses the fact that the radiance is measured from a single point in space. The single effective viewpoint property enables camera motion and scene structure estimation. Furthermore, images acquired by the sensor can be re-sampled to form other perspective image representations, which may be more suitable for certain tasks such as feature tracking or visualization.

The re-sampling into other perspective image representations requires that the sensor has a single effective viewpoint and that the intrinsic parameters of the sensor are known. Our sensor is based on a hyperboloid mirror to obtain panoramic vision. The alignment of the mirror with respect to the lens is critical to obtain a single effective viewpoint. We have described simple calibration schemes to correctly align the mirror and the lens, and to estimate the intrinsic parameters of the camera. Re-sampling furthermore requires a form of intensity interpolation. In the applications in forthcoming chapters we use the bi-linear interpolation method that may lead to aliasing in case the catadioptric image is under-sampled. A form of space variant filtering is required to resolve this issue. The issue has received little attention in panoramic vision literature and could be an interesting direction for future research.

Chapter 3

Appearance-based Robot Localization

3.1 Introduction

A mobile robot needs an internal representation of its workspace in order to localize itself. Localization is a prerequisite for optimal goal directed navigation in a large workspace. Internal sensors, such as shaft encoders, which measure the revolutions of the wheels, are useful to track the pose (location and orientation) of a robot as it moves. However, errors caused by wheel slippage accumulate. Therefore, the positional uncertainty can grow without bound when relying only on such "dead reckoning" for navigation. To counteract this effect, the robot has to observe the workspace with its external sensors (e.g. camera, laser range finder) and use a map in order to localize itself. The map specifies the relation between poses in the workspace and observations. Each observation provides (partial) evidence as to where the robot is located in the map. Localization now involves determining poses that "explain" the observation given the map.

Traditionally, mobile robots are equipped with an array of ultrasonic or infrared sensors that measure the distance to the nearest object in the workspace. The low-dimensional measurement vector obtained from such an array often carries little clues as to where a robot is located because the same sensor measurement profile may be obtained from many distinct poses in the workspace. Modern sensors, such as cameras, provide high-dimensional measurements that may yield more informative clues. Extracting relevant features from an image based on which different locations in the workspace can be distinguished is a crucial issue for fast global localization.

Many approaches have been proposed to extract a low dimensional feature vectors from high dimensional sensor data. The type of features that are extracted are related to the way in which the workspace is internally represented. Traditionally, range sensors are used and the map is typically represented as a 2-D or 3-D geometric map. Such maps range from relatively simple maps containing just the 2-D positions of (artificial or natural) landmarks to detailed 3-D CAD models. In *landmark-based* approaches, artificial or natural landmarks extracted from a novel sensor measurement are compared against those present in the map in order to infer the location. In *model matching* approaches, a local geometric model is extracted from a novel sensor measurement, which is subsequently matched against a global model to infer the location. Both model matching and landmark-based approaches rely crucially on the accurate and reliable extraction of salient features from the raw sensor data. A robot may be given a map of its workspace a priori, but a truly autonomous mobile robot should be capable of learning a map from sensor data acquired in the workspace. When the robot is equipped with a single vision sensor only, automatic learning of a 3-D geometric model for localization is complex. It requires reliable extraction of landmarks from images that can robustly be identified from various viewpoints. Part of the complexity arises due to the fact that the appearance of landmarks depends on the viewpoint from which they are perceived. Furthermore, in order to build such a map, the 3-D positions of the landmarks need to be known. The required 3-D information is not directly available, but rather needs to be inferred by triangulation of the landmark bearings as observed from different camera viewpoints.

As an alternative to learning a geometric map, one can attempt to model the relationship between images and poses directly. *Appearance modeling*, introduced by [61] in the context of object recognition, learns a model that relates poses to images observed at those poses from a set of training images that are labeled by their associated poses. Modeling in the high-dimensional space in which the images live is generally infeasible. Therefore, prior to modeling the dimensionality of images is reduced. Typically this is done by Principal Component Analysis (PCA). PCA finds a linear sub-space of the image space that preserves the directions in which the training images vary most. In order to perform localization on the basis of a novel image, an attempt is made to invert the learned relationship.

Appearance modeling has been adopted for robot localization by several researchers [26, 71, 42, 43, 41, 55]. A shortcoming in their localization approaches is that they ignore the fact that sensor measurements are inherently noisy so that at best a probabilistic estimate of the robot pose can be obtained. Furthermore, any prior belief that the robot already has about its pose is not incorporated in obtaining a pose estimate. These shortcomings can be overcome by explicitly representing and reasoning with uncertainty. In a probabilistic approach towards robot localization, a robot maintains a belief function (probability density function) over permissible poses in the workspace. Both sensing and acting affect the belief. A *motion model* predicts the uncertain result of an action and is used to update the belief accordingly. A probabilistic *observation model* relates poses to observations and is used to update the belief after obtaining a new observation.

In this chapter we present a method to learn an appearance-based observation model for probabilistic robot localization from a set of supervised training images. We propose a kernel density estimator (Parzen estimator) to represent the observation model. Similar to other works on appearance-based robot localization, we use PCA to reduce the dimensionality of images. PCA comes with a ranking of its features according to an image reconstruction criterion. We are however not interested in image reconstruction, but in robot localization. The ranking of individual PCA features according to a criterion related to the task of robot localization may be different. Such a criterion to characterize the performance of an observation model was recently proposed in [90]. We adopt this criterion to investigate whether the ranking of features that comes with PCA is also the best ranking for the task of robot localization.

This chapter is organized as follows. In section 3.2 we review related work on appearance-based robot localization and identify their limitations. Section 3.3 outlines a probabilistic framework for robot localization that we adopt to overcome these limitations. In section 3.4 we present our method to learn an appearance-based observation model for probabilistic localization. The model is experimentally evaluated in section 3.5. We concentrate on the problem of globally localizing a robot on the basis of a single observation assuming a uniform prior. We study how the performance depends on the parameters in our observation model. We investigate the number of PCA features needed for reliable localization, and we investigate whether the ranking of PCA features that comes with PCA is also optimal for robot localization. Finally, a discussion and conclusions are presented in section 3.6.

3.2 Appearance Modeling

Recently appearance-based approaches towards robot localization have been proposed. These approaches avoid the need to extract and match abstract features such as landmarks from high-dimensional sensor data. Instead, they attempt to model the sensor measurements as a function of the robot pose directly. Throughout our discussion we assume that sensor measurements are images, but the discussion is equally valid for other sensors such as laser range finders.

An image can be regarded as a vector in a high-dimensional space spanned by individual pixels. Typically, the set of images that are obtained in a particular environment form a subset of all possible images. For example, given that the robot's workspace is an office environment, it will be extremely unlikely that the robot perceives an image displaying a rock concert. Moreover, images acquired at nearby positions are often highly correlated. This suggests that the images that can be obtained in a particular environment live on a low-dimensional, but potentially highly curved and possibly self-intersecting manifold. Appearance-based modeling approaches aim to learn a representation of the manifold from a set of training images, where each training image is labeled by its associated pose.

Modeling the manifold in the high-dimensional image space is impractical because it requires a huge amount of training samples. Prior to modeling, appearance-based modeling approaches therefore first reduce the dimensionality of the images. An effective way to reduce the dimensionality of images, while preserving the directions in which the data varies most is Principal Component Analysis (PCA). PCA has been used for vision based robot localization [55, 71], visual servoing [12] and image synthesis of moving robot manipulators [39]. For dense range sensor scans PCA has been used to decrease the dimensionality of the data. [10, 104]. PCA calculates the eigenvectors of the covariance matrix of the set of training images. The eigenvectors with the largest corresponding eigenvalues are used to span an orthonormal basis of a low-dimensional subspace called the *eigenspace*. By projecting images into the eigenspace the major appearance characteristics, generally corresponding to low frequency signals present in the images, are retained. From a recognition point of view, the eigenspace has the attractive property that the distance between two points in eigenspace (projected images) is a least squares approximation to the correlation between the images from which the points are computed [62]. Correlation is a well established similarity measure. Several methods to calculate the eigenspace from a set of images are outlined in appendix A.

The projected training images yield a discrete sample of points on the appearance manifold. Several researchers fit an interpolating function through the training points in order to obtain a continuous representation of the appearance manifold that can be used for localization. In [71], multiple manifolds (each of which is indexed by the robot orientation) are modeled. A continuous representation of each manifold is obtained by linearly interpolating between training points. In their approach, localization on the basis of a novel image is done by first determining which of the manifolds is closest in order to estimate the robot orientation. Subsequently, the position of the robot is estimated by determining the pose associated with the closest point lying on the nearest manifold. In [41] cubic spline interpolation is used to obtain a continuous representation of the appearance manifold from cylindrical panoramic images, all of which are acquired under the same orientation. They also perform localization on the basis of a novel image by projecting the image onto the nearest point on the manifold. In [55] a nearest neighbor method is used for localization on the basis of a novel image.

A limitation shared by these approaches is that they give a *single* pose estimate. The estimate is based solely on the basis of a single observation. But what good is this estimate when we have no idea about its accuracy and reliability? Sensor measurements are inherently noisy, which implies that at best a probabilistic estimate of the robot pose can be obtained [91]. Furthermore, if a robot already roughly knows where it is, it would be preferable to utilize new observations to refine this knowledge, rather than to replace it. A robot localization framework that explicitly represents and reasons with uncertainty is discussed in the next section.

3.3 Probabilistic Robot Localization

In this section we outline a probabilistic framework for robot localization. In a probabilistic approach to robot localization the robot maintains a belief as to where it is located. Let $\mathbf{x}_t$ denote a random variable representing the state of the robot at time t. For localization, it is sufficient to characterize the state by the position and orientation with respect to a global frame of reference. Let us assume that at time t the robot obtains a sensor reading $\mathbf{z}_t$ from which it extracts a vector of features $\mathbf{y}_t$, which we will refer to as the observation at time t. At time t the robot executes an action $\mathbf{a}_t$ that terminates at time $t + 1$. The state of the robot $\mathbf{x}_t$ at time t cannot be observed directly, but rather has to be inferred from the sequence of past observations and actions.

Initially, at time $t = 0$ the robot has a prior belief as to where it is located. This belief is represented by a probability distribution $p(\mathbf{x}_0)$ that reflects the initial uncertainty. Observations and measurements change the robot's belief. Actions can be used to *predict* the next state. The robots belief after executing the $(t-1)$-th action will be denoted by $b_{t-1}(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1})$. Observations can be used to *correct* an estimate of the current state. The robots belief after obtaining the (t)-th observation will be denoted by $b_t(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1}, \mathbf{y}_t)$. We will treat these cases separately.

Correction. By application of the Bayes theorem

$$\begin{aligned} b_t(\mathbf{x}_t) &= p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1}, \mathbf{y}_t) && (3.1)\\ &= \alpha_t\, p(\mathbf{y}_t|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1}, \mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1}), && (3.2)\end{aligned}$$

where $\alpha_t = 1/p(\mathbf{y}_t|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1})$ is a normalizing constant that does not depend on the state. The Markov assumption states that sensor readings are conditionally independent of previous actions and observations given the current state;

$$p(\mathbf{y}_t|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1}, \mathbf{x}_t) = p(\mathbf{y}_t|\mathbf{x}_t). \tag{3.3}$$

By application of the Markov assumption, equation 3.1 reduces to

$$b_t(\mathbf{x}_t) = \alpha_t p(\mathbf{y}_t|\mathbf{x}_t) b_{t-1}(\mathbf{x}_t). \tag{3.4}$$

We see that the most recently available estimate of the robot pose, $b_{t-1}(\mathbf{x}_t)$ is corrected by the probability density function $p(\mathbf{y}_t|\mathbf{x}_t)$ of the expected observation from a given location by Bayesian inversion. The density $p(\mathbf{y}_t|\mathbf{x}_t)$ is called the *observation model.* The observation model relates sensor measurements to locations of the environment and can thus be regarded as a map of the environment. Different kinds of maps have been proposed in literature. The map may describe explicit properties of the environment such as the positions of such as positions of landmarks [92] or occupancy values [48]. Alternatively, the map may be an implicit model directly relating sensor patterns and robot poses such as neural networks [66], radial basis functions [103] or look-up tables [10].

Prediction. The effect of actions is that they change the pose of the robot and thus its belief. Using the theorem of total probability

$$\begin{aligned} b_{t-1}(\mathbf{x}_t) &= p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1}) && (3.5)\\ &= \int p(\mathbf{x}_t|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1}, \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_1, \mathbf{a}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{a}_{t-1}) d\mathbf{x}_{t-1}. && (3.6)\end{aligned}$$

The state $\mathbf{x}_{t-1}$ does not depend on the action $\mathbf{a}_{t-1}$ executed from there. Furthermore, exploiting the Markov assumption again, the state $\mathbf{x}_t$ only depends on the state $\mathbf{x}_{t-1}$ and the action $\mathbf{a}_{t-1}$. Equation 3.5 then reduces to:

$$b_{t-1}(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{a}_{t-1}, \mathbf{x}_{t-1}) b_{t-1}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}. \tag{3.7}$$

We see that the prediction phase involves a convolution of the belief $b_{t-1}(\mathbf{x}_{t-1})$ with the density $p(\mathbf{x}_t|\mathbf{a}_{t-1}, \mathbf{x}_{t-1})$, which is referred to as the *motion model.* The motion model describes the effect that a motion command $\mathbf{a}$ has on the location of the robot and can be seen as a generalization of mobile robot kinematics.

In literature on Markov localization, often simple kinematic models are used and Gaussian noise is added to the prediction to account for model limitations and wheel slippage. These simple approaches tend to give an overly pessimistic estimate of the uncertainty region resulting from a motion command. If odometry would be the only source of information a robot could rely on for localization, it would certainly pay off to use an accurate, calibrated model of odometry and its error propagation instead. The main practical motivation for using simple models is that pose estimates will be corrected by external sensing in any case.

By combining equations 3.4 and 3.7 we arrive at the following recursive for probabilistic robot localization:

$$b_t(\mathbf{x}_t) = \alpha_t p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{a}_{t-1}, \mathbf{x}_{t-1}) b_{t-1}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}. \tag{3.8}$$

In order to implement equation 3.8 we need to specify a motion model, an observation model, and a representation of the belief function that can accurately and efficiently be maintained over time. Inventing a motion model is relatively easy as it follows from the robot kinematic model. Various well established belief function representations and methods to maintain them have been described in literature. They are briefly outlined in appendix B. In this chapter, we focus on the observation model. The availability of a good observation model is crucial for fast global localization. We address the problem of learning an appearance-based observation model from images labeled by their respective pose. This problem has received little attention in robotics literature.

3.4 Our Observation Model

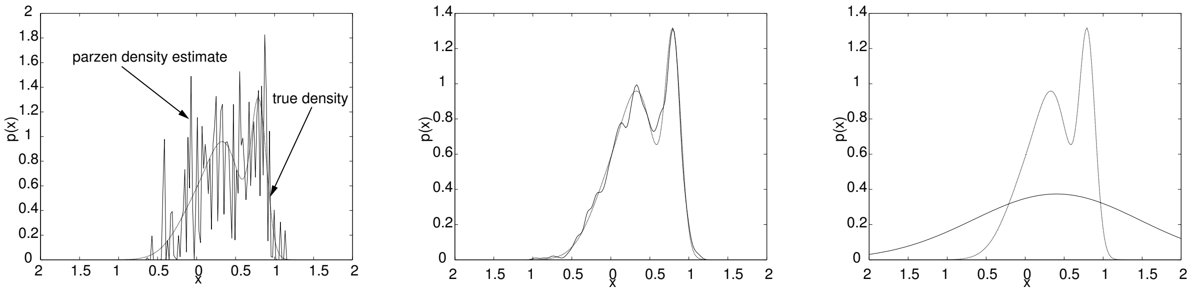
In this section we present our method to learn an appearance-based observation model $p(\mathbf{y}|\mathbf{x})$ for probabilistic robot localization from a set of N training samples

$$\{(\mathbf{z}_1, \mathbf{x}_1), (\mathbf{z}_2, \mathbf{x}_2), \ldots, (\mathbf{z}_N, \mathbf{x}_N)\},$$

where $\mathbf{z}_i$ is a sensor measurement and $\mathbf{x}_i$ is the pose at which the measurement was taken. In our specific application, the sensor measurements $\mathbf{z}$ are cylindrical panoramic images, and we use PCA to extract observations $\mathbf{y}$ from the images. Note however that our approach is not restricted to these specific types of sensor measurements and features.

3.4.1 Model representation

We propose to model $p(\mathbf{y}|\mathbf{x})$ by a kernel density estimator, also known as the Parzen estimator [70]. The Parzen estimator approximates the density by a normalized sum of



(a) too small

(b) reasonable

(c) too large

Figure 3.1: Illustration of Parzen density estimation. A set of 500 samples are drawn from the true distribution (dotted curve). The plots show the effect on the Parzen density estimate (solid curve) of varying kernel widths. a) a too small kernel width ($h = 0.001$) results in an over-fitted estimate, b) a reasonable approximation ($h = 0.04$), c) a too large kernel width ($h = 1$) results in an overly smooth estimate.

identically shaped kernel functions centered at the training samples. Let $\mathbf{v}_1, \ldots, \mathbf{v}_N$ be a set of N d-dimensional training samples. The Parzen density estimate at a point $\mathbf{v}$ is given by:

$$p(\mathbf{v}) = \frac{1}{N} \sum_{n=1}^{N} K(\frac{\mathbf{v} - \mathbf{v}_i}{h}), \tag{3.9}$$

where K is an arbitrary bounded probability density and h is the width of the kernel, which controls the smoothness of the density estimate.

The Parzen estimator is illustrated in figure 3.1. The dotted function represents an underlying 1-D distribution from which samples were drawn. The solid functions represent the Parzen density estimates for various kernel widths. In figure 3.1a the kernel width is chosen to be very small, resulting in a wildly fluctuating approximation of the true underlying density. In figure 3.1b an appropriate kernel width was used and the Parzen estimate gives a reasonably close approximation of the underlying density. Finally, figure 3.1c shows what happens when the kernel width is chosen to large. In this case, the density estimate is too smooth and resembles a single Gaussian with a large variance centered at the data mean.

The designer of the Parzen estimator has the freedom to choose the kernel shape and has to determine the kernel width h. In this work we use Gaussian kernels. A multivariate d-dimensional Gaussian kernel with width h and centered at a point $\boldsymbol{\mu}$ is given by

$$g(\mathbf{v} - \boldsymbol{\mu}) = \frac{1}{(2\pi)^{d/2} h^d |\Sigma|^{1/2}} \exp[-\frac{1}{2h^2}(\mathbf{v} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{v} - \boldsymbol{\mu})]. \tag{3.10}$$

In order to construct a kernel density estimator for the observation model $p(\mathbf{y}|\mathbf{x})$ (see equation 3.4), we first express it as

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{y},\mathbf{x})}{p(\mathbf{x})}. \tag{3.11}$$

Both the joint density $p(\mathbf{y},\mathbf{x})$ and the density $p(\mathbf{x})$ are then modeled by a Parzen density estimator. In order to model the joint density $p(\mathbf{y},\mathbf{x})$ we assume that $\mathbf{x}$ and $\mathbf{y}$ are *locally* independent. That is, we assume that $\mathbf{x}$ and $\mathbf{y}$ are independent around the training samples. We express $p(\mathbf{x}|\mathbf{y})$ as

$$p(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{n=1}^{N} g_y(\mathbf{y}-\mathbf{y}_n)g_x(\mathbf{x}-\mathbf{x}_n), \tag{3.12}$$

where $g_y()$ and $g_x()$ represent univariate kernels. Similarly, we model $p(\mathbf{x})$ as

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} g_x(\mathbf{x}-\mathbf{x}_n). \tag{3.13}$$

The set of samples from which the observation model is estimated consists of feature vectors $\mathbf{y}$ and their associated poses. In our data set, the training poses are points on a two-dimensional grid in world space where the size of a grid cell is $c \times c$. We therefore choose to use a univariate Gaussian for the $\mathbf{x}$-kernels. We fix the kernel width of the $\mathbf{x}$-kernels at $h_x = c/2$, i.e. half the separating distance between two neighboring points on the grid. We have no a priori knowledge about the density of the feature vectors in the q-dimensional feature space. In lack of such knowledge, we assume that the features are uncorrelated, so that the covariance matrix in equation 3.10 becomes diagonal. Each $\mathbf{y}$-kernel is thus represented by a univariate q-dimensional Gaussian. What is left to estimate is the optimal kernel width of the $\mathbf{y}$-kernels. In order to quantify what optimal is, we need a criterion to characterize the performance of our observation model. We address the issue of estimating the optimal kernel width in section 3.4.3. Before we do so, we first derive an evaluation criterion in the following section.

3.4.2 Model evaluation

An obvious measure to characterize the performance of an observation model is the expected deviation between estimated and true locations, as was recently proposed in [89]. Let $\mathbf{x}^*$ denote the true pose of the robot, and let $L(\mathbf{x}^*,\mathbf{x})$ denote a loss function measuring the error between the true pose $\mathbf{x}^*$ and an arbitrary other pose $\mathbf{x}$. After observing $\mathbf{y}^*$, the Bayesian localization error at $\mathbf{x}^*$ is obtained by integrating the error L over all possible poses $\mathbf{x}$ weighted by the likelihood $p(\mathbf{x}|\mathbf{y}^*)$ that the robot assigns to them, giving

$$\varepsilon(\mathbf{x}^*, \mathbf{y}^*) = \int_{\mathbf{x}} L(\mathbf{x}^*, \mathbf{x}) p(\mathbf{x}|\mathbf{y}^*) d\mathbf{x} \tag{3.14}$$

$$= \int_{\mathbf{x}} L(\mathbf{x}^*, \mathbf{x}) \frac{p(\mathbf{y}^*|\mathbf{x}) p(\mathbf{x})}{p(\mathbf{y}^*)} d\mathbf{x} \tag{3.15}$$

$$= \int_{\mathbf{x}} L(\mathbf{x}^*, \mathbf{x}) \frac{p(\mathbf{y}^*|\mathbf{x}) p(\mathbf{x})}{\int_{\mathbf{x}} p(\mathbf{y}^*|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}} d\mathbf{x}. \tag{3.16}$$

Thus far, the error ε corresponds to a single pose and observation only. By averaging over all possible observation $\mathbf{y}^*$ obtained from $\mathbf{x}^*$ and all possible poses $\mathbf{x}^*$, weighted by the likelihood of occurrence $p(\mathbf{x}^*, \mathbf{y}^*)$, the average Bayesian localization error is derived as

$$E = \int_{\mathbf{x}^*} \int_{\mathbf{y}^*} \varepsilon(\mathbf{x}^*, \mathbf{y}^*) p(\mathbf{y}^*, \mathbf{x}^*) d\mathbf{y}^* d\mathbf{x}^* \tag{3.17}$$

$$= \int_{\mathbf{x}^*} \int_{\mathbf{y}^*} \varepsilon(\mathbf{x}^*, \mathbf{y}^*) p(\mathbf{y}^*|\mathbf{x}^*) p(\mathbf{x}^*) d\mathbf{y}^* d\mathbf{x}^* \tag{3.18}$$

$$= \int_{\mathbf{x}^*} \int_{\mathbf{x}} L(\mathbf{x}^*, \mathbf{x}) p(\mathbf{x}^*) p(\mathbf{x}) \int_{\mathbf{y}^*} \frac{p(\mathbf{y}^*|\mathbf{x}^*) p(\mathbf{y}^*|\mathbf{x})}{\int_{\mathbf{x}} p(\mathbf{y}^*|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}} d\mathbf{y}^* d\mathbf{x} d\mathbf{x}^*. \tag{3.19}$$

Equation 3.17 measures the true average Bayesian localization error. Assuming that the training data were sampled from uniform priors $p(\mathbf{x})$ and $p(\mathbf{x}^*)$, the *empirical* average Bayesian localization error can be approximated by Monte Carlo estimation as

$$E \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} L(\mathbf{x}_i, \mathbf{x}_j) \sum_{k=1}^{N} \frac{p(\mathbf{y}_k|\mathbf{x}_i) p(\mathbf{y}_k|\mathbf{x}_j)}{\sum_{l=1}^{N} p(\mathbf{y}_k|\mathbf{x}_l)}. \tag{3.20}$$

The above empirical error measure converges to the true error measure as the size of the data set grows to infinity. Although in practice the size of the training set is always limited, the empirical error measure allows us to evaluate and compare the performance of different observation models $p(\mathbf{y}|\mathbf{x})$. The smaller the value of E is, the more reliable the global localization is.

3.4.3 Model estimation

The average Bayesian localization error presented in the previous section provides us a handle to determine the optimal kernel width for our observation model; the optimal kernel width is the one that minimizes the average Bayesian localization error. It is obvious that the optimal kernel width for the set of training samples is zero. An evident drawback of zero kernel width is that the resulting observation model lacks any generalizing capabilities. To overcome this undesirable situation one can use a cross-validation method to estimate the kernel width from the training data only [15]. In cross-validation, for each

$i \in 1, \dots, N$ a leave-one-out estimate is formed by omitting the contribution of the i-th sample in the estimation of the density at that point and determine the optimal kernel width by minimizing the Bayesian localization error at the i-th datum as a function of the kernel width h. Doing so for every sample in the training set, an estimate of h may be formed by taking the average over all estimate kernel widths. Excluding the sample causes a bias towards larger kernel widths because neighboring points have to account for the density at the point. Alternatively, the optimal kernel width can be found by minimizing the Bayesian localization error on an independent test set as a function of the kernel width. We adopt the latter approach.

The Bayesian localization error is minimized using Golden section search with Brent's parabolic interpolation [73]. Golden section search is a general method for 1-D function minimization. A minimum of an arbitrary 1-D function is said to be bracketed by a triplet of points (a, b, c), where $a < b < c$, such that $f(b) < f(a)$ and $f(b) < f(c)$. In order to hunt down the minimum given a bracketing triplet, a new point x that lies either in the interval (a, b) or in the interval (b, c) can be chosen. Suppose that $a < x < b$ is chosen. If $f(x) > f(b)$, a new bracketing triplet can be formed as (x, b, c). If $f(x) < f(b)$, a new bracketing triplet can be formed as (a, x, b). In any case, the middle point of the new bracketing triplet always corresponds to the lowest function value found so far. The procedure can be repeated using the new triplet until convergence. Golden section search gives the optimal recipe for choosing the next point x to be tried. It turns out that, for arbitrary 1-D functions, the best strategy is to choose the next point x a fixed fraction (originating from a so-called golden section) into the larger of the two intervals. Brent's method speeds up the convergence by switching to parabolic interpolation if the function to be minimized has a parabolic shape in the vicinity of its minimum.

3.5 Experiments

We have described a method to learn an appearance-based observation model for probabilistic robot localization. In a series of experiments we investigate 1) how the performance of the model depends on the model parameters, 2) how many features are needed for global localization, and 3) which PCA features yield the best performance. We consider a scenario in which the robot has to localize itself globally on the basis of a single observation where a uniform prior is assumed.

3.5.1 Data-sets and pre-processing

We evaluate our observation model on real image data obtained by a catadioptric vision sensor mounted on top of a mobile robot. The image data was kindly provided by Tsukuba Research Center (Japan), Real World Computing Program, in the form of the MEMORABLE robot database. The database contains sensor data (laser range scan, infrared sensors, bumpers and catadioptric vision) obtained at more than 8000 poses

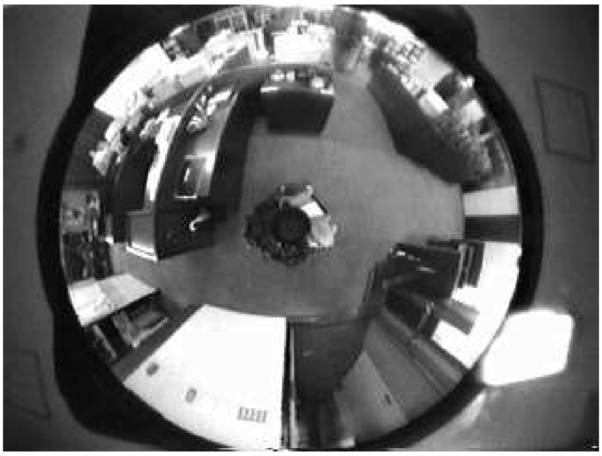


Figure 3.2 : A catadioptric image from the TRC Database.

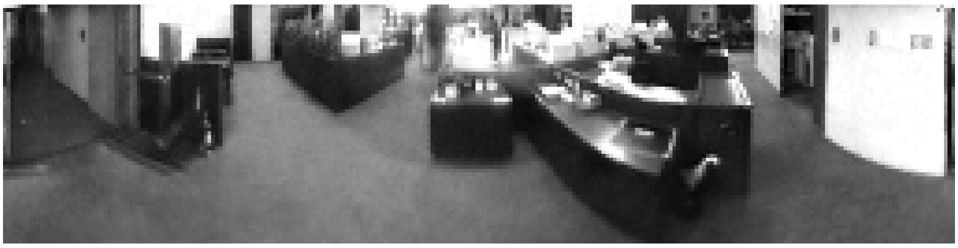


Figure 3.3: A cylindrical panoramic image derived from the catadioptric image displayed in figure 3.2.

in a 17 × 17 m office environment. The data was obtained by accurately positioning a Nomad 200 robot at discrete coordinates on a virtual sampling grid over the workspace. Each cell of the virtual grid is 10 × 10 cm. A 2-D map of the workspace is displayed in figure 3.4. The head of the Nomad 200 always faces the same direction so that the pose of the robot may be characterized by the position only. Figure 3.2 displays an image from the database. We transform the catadioptric images in the database to 720 × 200 pixels cylindrical panoramic images using the methods presented in chapter 2. The cylindrical panoramic images are smoothed by a Gaussian function and subsequently sub-sampled to a resolution of 64 × 256 pixels to reduce the dimensionality. An example is shown in figure 3.3. The resulting images are then normalized so that the sum of their squared pixel intensity values equals one. This simple normalization renders the image representation less sensitive to global intensity changes. A set of 2000 images was randomly selected from the database to derive the eigenvectors and associated eigenvalues. The first 5 eigenvectors of the database are displayed in figure 3.5.

3.5.2 Estimation of the kernel width

Estimation of the parameters of our observation model requires a set of training samples. The MEMORABLE database contains over 8000 samples, which in principle can all be

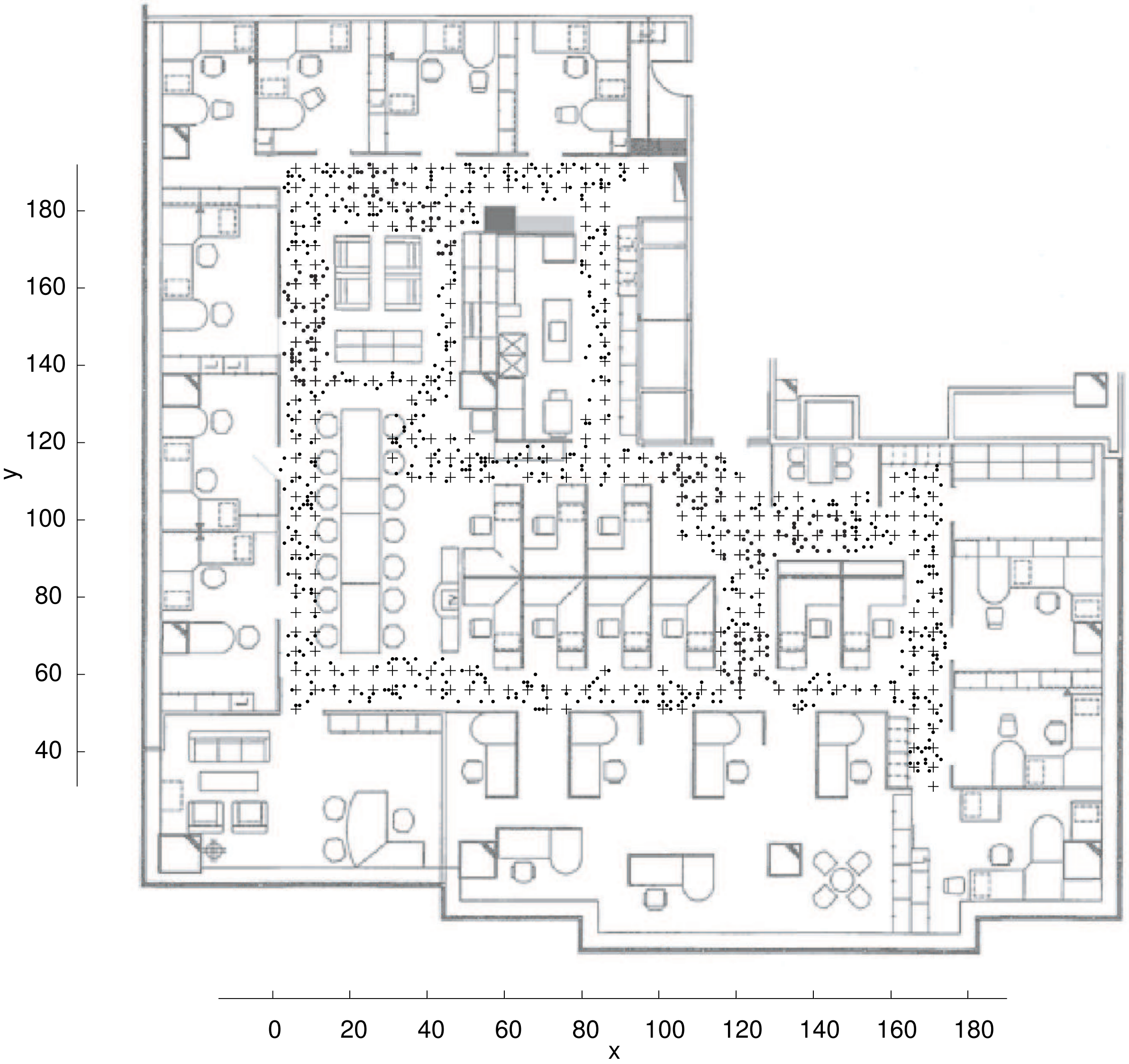


Figure 3.4: The environment from which the images were taken. Positions marked by a + indicate where images that were used to estimate our observation model were obtained. Position marked by a · indicate positions that were used to evaluate the performance of our observation model. The units of the axes are 10^{-1} m.

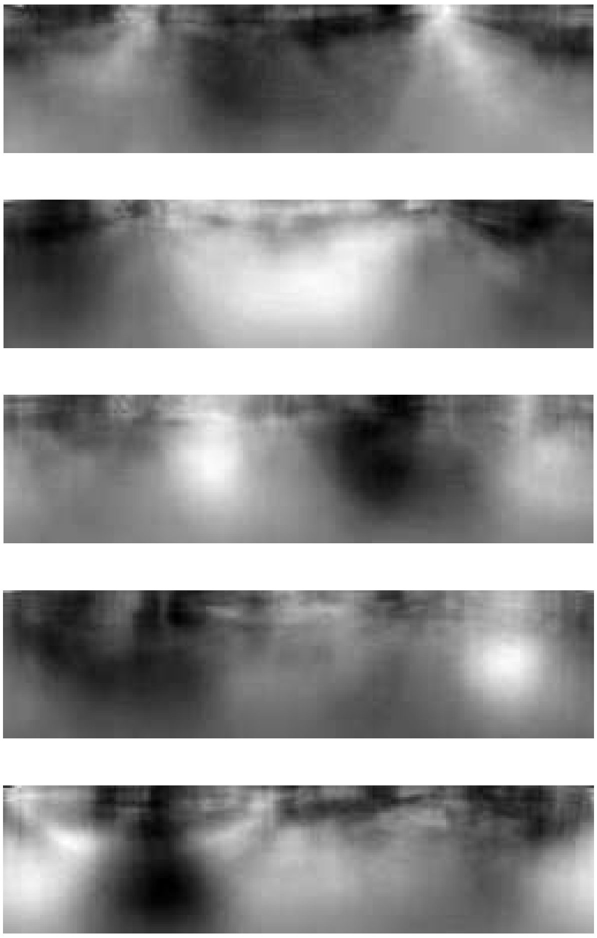


Figure 3.5 : The five most prominent eigenvectors.

used to construct the observation model. However, in the operational localization stage distances from all possible poses and their associated feature vectors (either represented by discrete cells as in grid-based methods or by particles as in particle-based methods) to the novel observation have to be calculated. Distance computations are computationally demanding.

We therefore limit the number of training samples, and use a subset of 303 samples as training samples. The training samples are obtained at discrete positions of a grid, each cell of which is 50×50 cm. Another (non-overlapping) subset of 400 samples is uniformly drawn from the poses present in the database for estimating the optimal kernel width.

We wish to investigate the performance of the observation model for different individual PCA features and for PCA feature vectors of increasing dimensionality. In our experiments we use the first 25 most prominent eigenvectors as features. We instantiated 50 observation models, each of which uses a different subset of PCA features. The first 25 models use a single PCA feature. The other 25 use PCA feature vectors of increasing dimensionality, i.e. the first model uses the first PCA feature, the second model uses the first and second PCA feature etc. For each model we estimate the optimal kernel width.

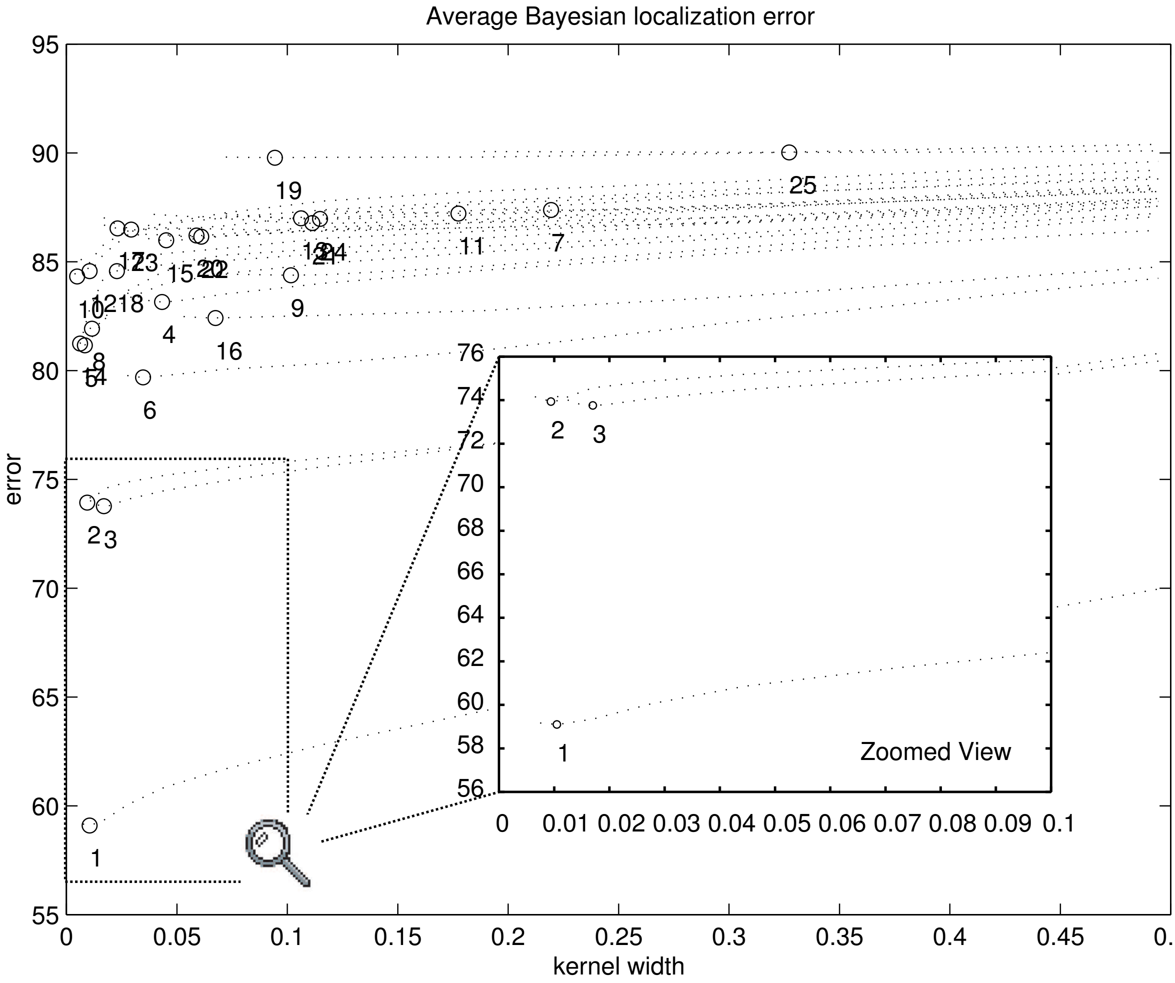


Figure 3.6: The average Bayesian localization error (in 10^{-1} m) plotted as a function of the kernel width h for individual features. The optimal kernel width for each model found by the minimization procedure is marked by a $\circ$.

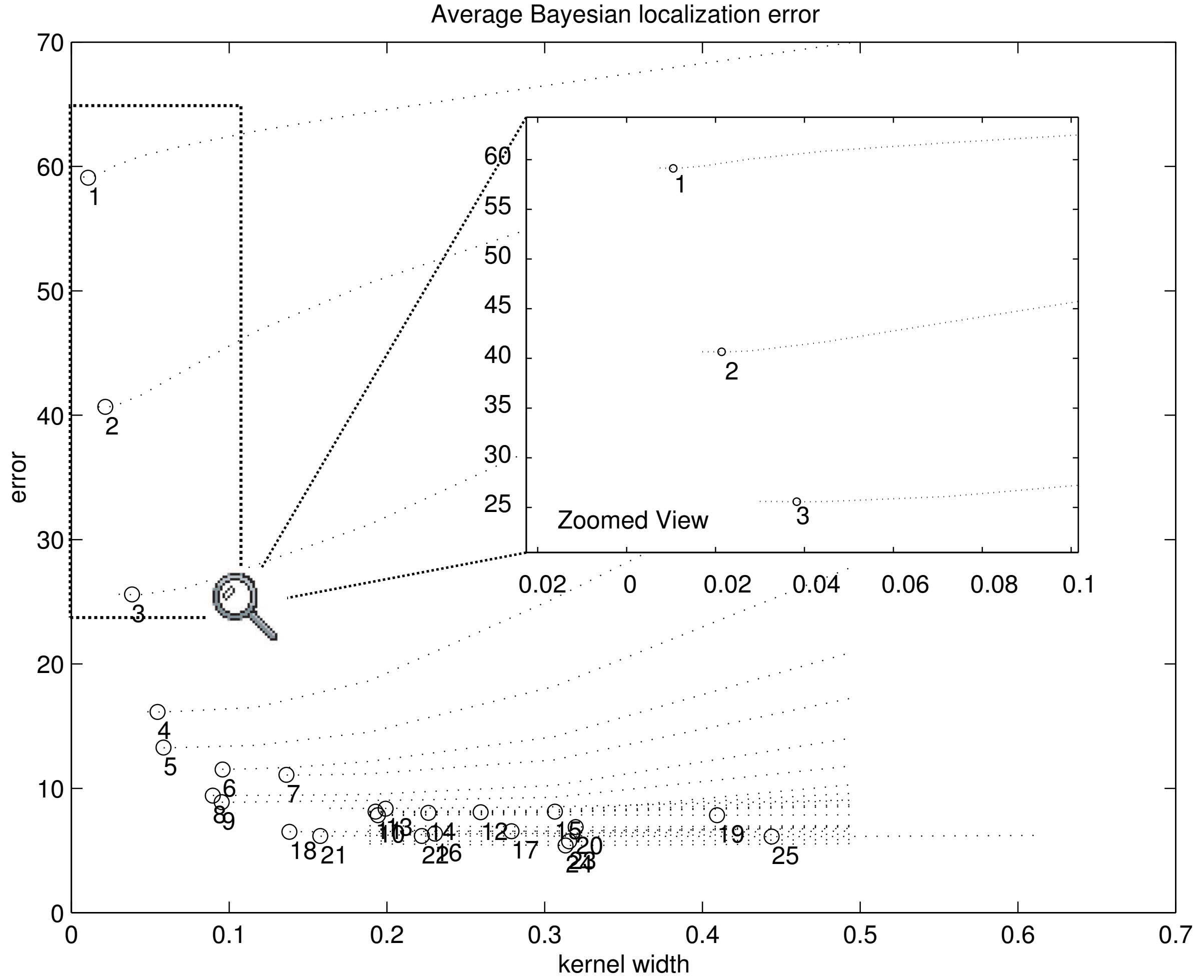


Figure 3.7: The average Bayesian localization error (in 10^{-1} m) plotted as a function of the kernel width h for feature vectors of increasing dimensionality.The optimal kernel width for each model found by the minimization procedure is marked by a $\circ$.

The resulting optimal kernel width for the different models are displayed in figures 3.6 and 3.7. Figure 3.6 displays the average Bayesian localization error evaluated on the training set for the models using a single PCA feature as a function of the kernel width h. The optimal kernel width for each model as found by the minimization procedure is marked by a $\circ$. The numbers below the marks identify the model. The n-th model is uses the n-th PCA feature according to the ranking that comes with PCA. Figure 3.7 displays a similar plot for the models trained on vectors of PCA features. In this plot, the n-th curve corresponds to the model trained on the first n PCA features according to the ranking of features that comes with PCA. Note that the all curves start at a kernel width larger than zero. The reason is that if the kernel width is chosen to tight, the Parzen density estimate becomes zero at points other than the training samples.

We observe that for the single feature models a rather well defined minimum exists for the first few PCA features. For PCA features with a larger index, the minimum is less well defined. This could be expected, as the features with a lower index are least sensitive to noise in the original images. For the models using feature vectors of increasing dimensionality, we observe that a rather well defined minimum exists for low dimensional feature vectors (up to 3 components). For higher dimensional feature vectors the minimum is less well defined. This implies that performance of the observation model is less sensitive to the choice of kernel width. The plot also shows that the optimal kernel width increases with increasing feature vector dimensionality. This may be explained by the fact that data tends to become sparse in high-dimensional spaces and is related to the distribution of the data. Finally, we note that the average Bayesian localization error on the training data decreases when more PCA features are taken into account.

3.5.3 Performance of the observation models

The performance of each observation model is evaluated on a validation set. The validation contains samples uniformly drawn from the admissible poses. The validation set is disjoint from the set of samples used to estimate the observation model (both training samples, as well as the samples used to estimate the kernel width). Figure 3.4 displays the training points (marked by squares) and the validation points (marked by dots).

Performance of single feature models. In this experiment we investigate whether the ranking of features that comes with PCA is also optimal for robot localization. In order to do so we evaluate the performance of the models trained on single features on the independent validation set. Figure 3.8 displays the results. The plot shows the average Bayesian localization error (in 10^{-1} m) as a function of the PCA feature index. The general trend of the curve indicates that performance decreases with increasing PCA feature index, as we also saw during training. This indicates that the features most important for reconstruction are also the ones most important for global localization. Note that an upper bound on the localization error exists due to a finite size of the workspace represented by the training samples.

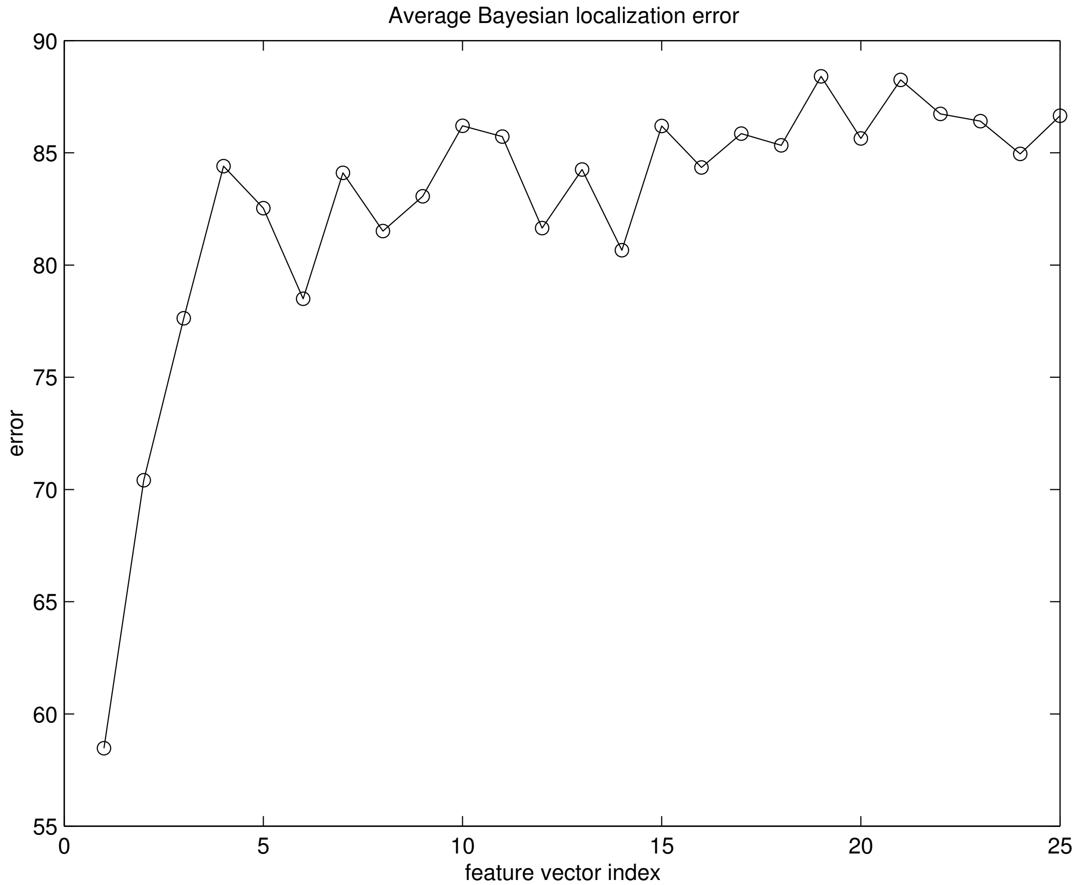


Figure 3.8: The average Bayesian localization error (in 10^{-1} m) obtained on a test set for single PCA features.

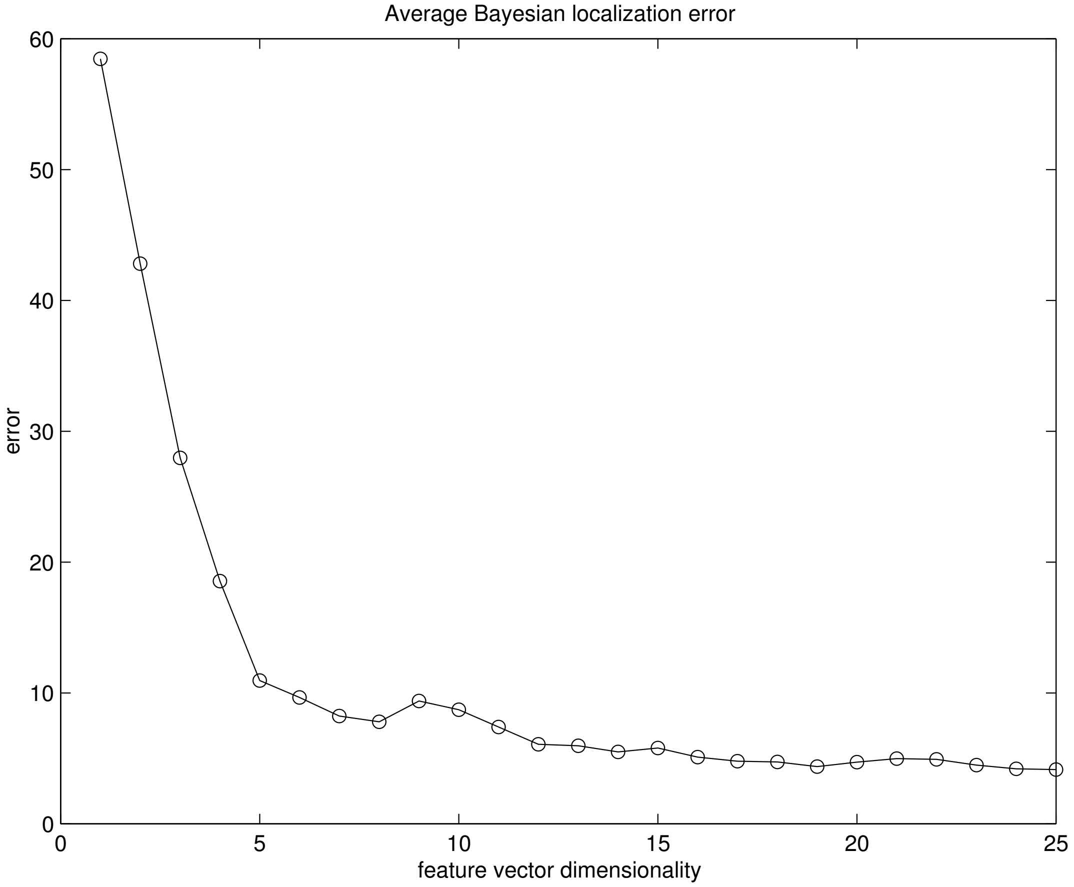


Figure 3.9: The average Bayesian localization error (in 10^{-1} m) on a test set for feature vectors of increasing dimensionality.

Performance of multiple feature models. The previous experiment showed that PCA features with lower indices generally yield a better performance than PCA features with higher indices. Still, individual features have relatively little explanatory power. By combining individual features into feature vectors we expect that a better performance can be achieved. In the current experiment we use the ranking of features that comes with PCA and investigate the number of PCA features needed for localization. In order to do so, we evaluate the performance of the observation models using feature vectors of increasing dimensionality on the independent evaluation set. The results are displayed in figure 3.9. The plot displays the average Bayesian localization error as a function of the PCA feature vector dimensionality. We see that by using more features the localization error decreases. The increase of performance is most significant for up to 10-dimensional feature vectors. For 10 and more dimensions, little is gained. Faced with the tradeoff between computational demands and accuracy, we would use about 10 features to perform localization, giving an estimated average Bayesian localization error of about $0.7m$ in an workspace of about $17m \times 17m$.

In order to achieve a better performance, the number of training samples used to represent the observation model could be increased. Unfortunately, this results in much higher computational demands. The localization procedure spends most of its computation time in calculating distances. Localization requires calculation of distances between each possible pose and each training pose (KN distances in a 2-D or 3-D space), and calculation of distances between an observation and the observations associated with each training pose (K distances in the feature space).

3.5.4 A concrete localization example

The average Bayesian localization error may give a slightly pessimistic view of the accuracy of localization because it may be affected strongly by a few outliers. Some regions in the workspace may appear very similar (particularly when low-dimensional feature vectors are used). An example is shown in figure 3.10. The top image corresponds to the image obtained at position (5,180). Its 20-D projection (back-projected to the image domain) is shown in the middle. The maximum a posteriori position estimate obtained using the observation model and the 20-D feature vector corresponds to location (116,116). The image obtained at (116,116) is shown at the bottom.

In the process of experimenting with the observation models, we observed that the Bayesian localization error for a single observation is often far below the reported best average of 0.7 m. For illustration, we display the a posteriori position estimates obtained for a 1-,2-,4- and 7-dimensional feature vector from an image obtained at position (69,51). Figure 3.11 displays the posterior density obtained using the first PCA feature only. The figure displays the posterior density over the allowable poses in the workspace (i.e. those present in the database). The density is estimated at the training samples (marked by dots). Linear interpolation was used to approximate the density estimates at all allowable poses in the workspace (i.e. those present in the whole database). The likelihood at each allowable pose is represented by a gray value. The gray values are scaled from zero

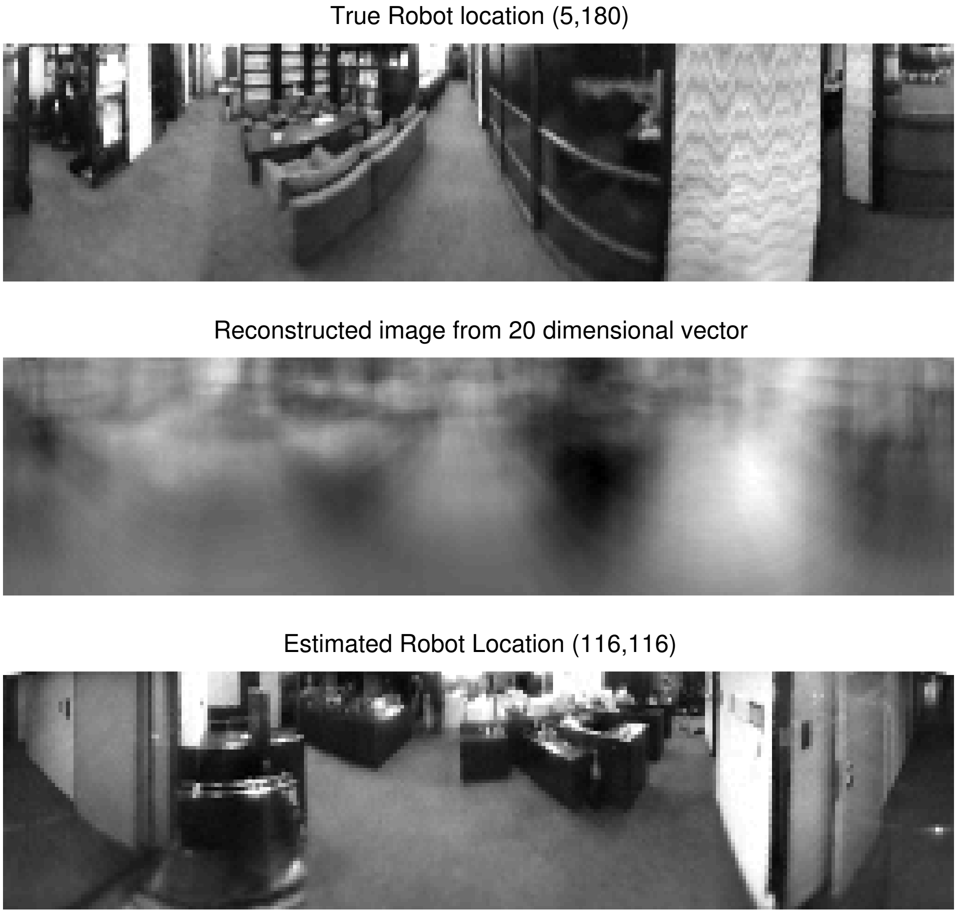
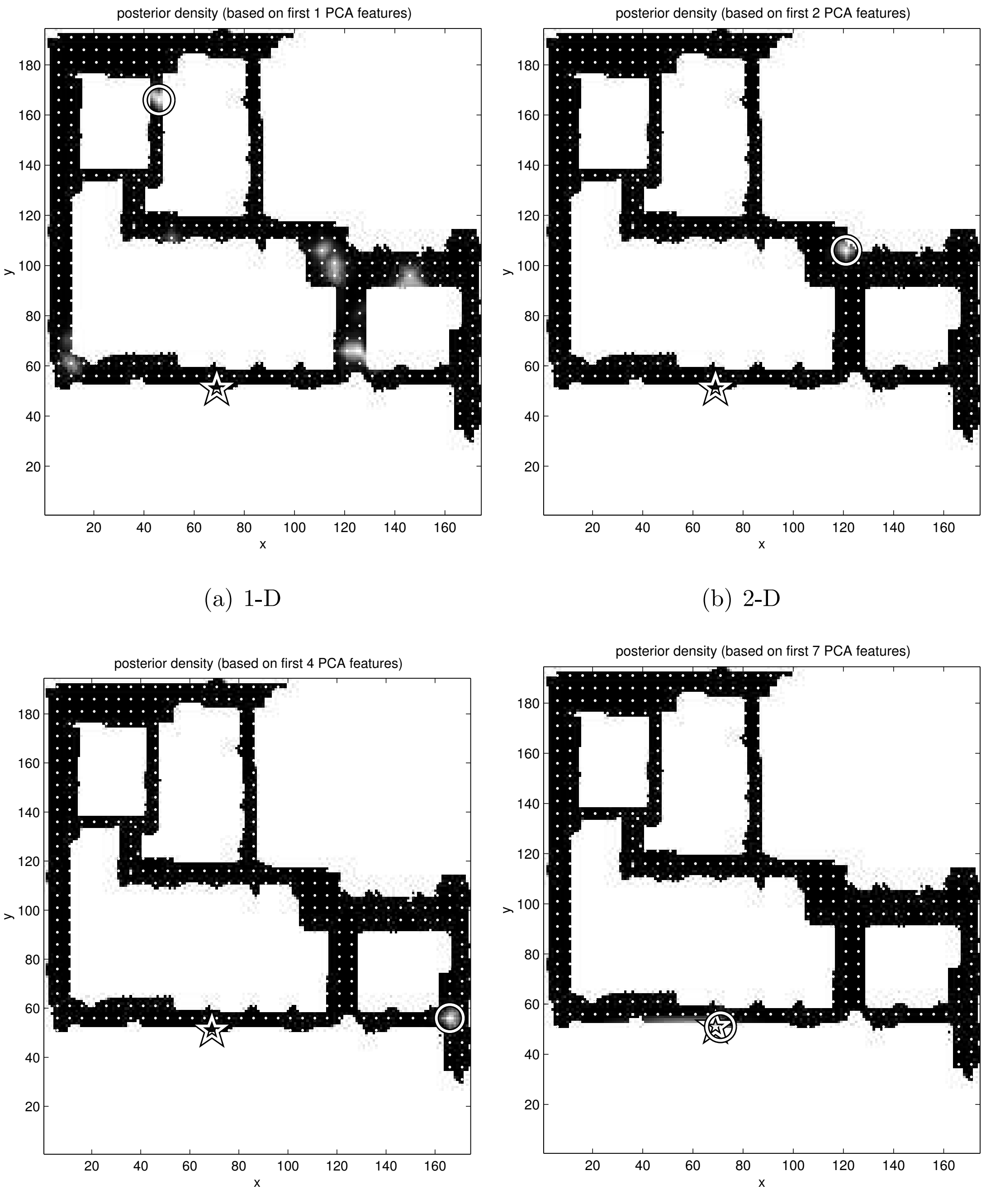

True Robot location (5,180)
Reconstructed image from 20 dimensional vector
Estimated Robot Location (116,116)

Figure 3.10: top) image at test location, middle) its representation in feature space, bottom) image from maximum likelihood posterior estimate

likelihood (black) to maximum likelihood (white). The true pose is marked by a star. The maximum likelihood posterior pose estimate is marked by a circle. For the single PCA feature, the density is multi-modal. The 2-D feature vector results in a uni-modal estimate, which is peaked at the wrong location. The peak remains at the same location up to a 4-D feature vector. The 4-D feature vector results in another uni-modal distribution, albeit again at an incorrect position. Extending the feature vector dimensionality, we observe that if the dimensionality exceeds 7, the distribution is uni-modal and peaked at the true robot location.

3.6 Discussion and Conclusions

Appearance modeling is an attractive method to construct a model for robot localization because of its simplicity; no reliable landmark extraction and identification is needed. The primary contribution of this chapter are the development and evaluation of a probabilistic appearance-based observation model for global localization.


posterior density (based on first 1 PCA features)

posterior density (based on first 2 PCA features)

posterior density (based on first 4 PCA features)

posterior density (based on first 7 PCA features)

(a) 1-D

(b) 2-D

(c) 4-D

(d) 7-D

Figure 3.11: Posterior distributions. Figure a) displays the posterior distribution after obtaining a 1-D PCA feature from the test position marked by the star. The small white dots indicate the training poses. The distribution is displayed as a gray map, where black corresponds to zero likelihood and white corresponds to the maximal likelihood. The maximum likelihood pose estimate is indicated by a circle. The other plots shown have a similar interpretation but differ in the number of features used to localize, b) 2 features, c) 4 features, and d) 7 features.

We have presented a method to estimate an appearance-based observation model from a set of training images, which are labeled by their respective poses. The performance of our observation model was optimized by minimizing the average Bayesian localization error [89]. We applied the same criterion for model selection purposes.

Our experimental study is limited to the problem of globally localizing the robot on the basis of a single observation, assuming an uninformed prior over the admissible poses in the workspace. We showed that our appearance-based observation model can be used to localize a mobile robot in an office environment of about 17 × 17 m. Localization accuracy improves if the number of PCA features used increases. On our database we established an average Bayesian localization error of 0.7 m when using 16 PCA features and construct the observation model from 300 training samples.

A disadvantage of our appearance-based method is that the method is sensitive to illumination changes and occlusions. In principle, illumination changes and occlusions can be accounted for by simply incorporating such circumstances in the set of training data used to represent the observation model. In practice, gathering images under all possible circumstances is impossible. Furthermore, the huge resulting observation model hampers real-time localization. A more pragmatic approach to deal with illumination changes could involve illumination invariant image representations (such as hue- or edge density images) instead of intensity images used in this work. Occlusions could be dealt with by robust PCA methods [51, 42] or by using sub-images taken from the panoramic view and combining pose estimates obtained on the basis of each sub-image in a principled manner [69].

We have experimentally investigated whether the ranking of individual PCA features — which is optimal for reconstruction of the images from which they are derived — is also optimal for localization. On our database this indeed is the case under the global localization scenario; the eigenvectors with the largest corresponding eigenvalues are most important for global localization. In situations where the robot's belief is peaked at multiple distinct poses (because the images that can be observed from those poses have a similar global appearance), other PCA features may become more important in order to disambiguate the belief. It would be interesting to investigate feature selection for such "situated" scenarios.

The average Bayesian localization may be strongly be influenced by a few images in the test set whose appearance is very similar, but whose associated poses are separated by a large distance. Although we consider the criterion to be useful for model estimation and model selection purposes, it gives a bit of a pessimistic view of the achievable localization accuracy. During experimentation, we have found that when the posterior belief is unimodal, it is usually peaked close to the true robot pose (provided a sufficient number of PCA features are used). Such quick localization is a prerequisite for optimal goal directed navigation.

Chapter 4

Panoramic Stereo Vision

4.1 Introduction

The ability to acquire knowledge about the 3-D structure of a workspace has many applications in mobile robotics. Such knowledge could for instance be used to construct a metric map of the environment or to avoid collisions with objects in the workspace. While range sensors such as ultrasonic and laser range finders provide depth information directly, it is also possible to estimate depth from images obtained from different nearby viewpoints. A traditional approach to obtain a 3-D reconstruction of a scene from images is stereo vision [17]. Stereo vision is the process of recovering depth information from two (or more) images obtained by a calibrated camera placed at different but known poses. The fundamental problem in stereo vision is establishing correspondences between points in the images that are the projections of the same physical point. This problem is known as the *correspondence problem*. Once the correspondences are known, the 3-D coordinates of the point relative to a chosen frame of reference can be reconstructed using triangulation. This is known as the *reconstruction problem.*

The key assumption in the correspondence problem is that the projection of a local neighborhood in the world gives rise to similar intensity patterns in images obtained at different camera poses. Given a window centered at a point in one image, a correspondence can be found by scanning the other image(s) with a search window to find the position of the search window that yields the maximal similarity. The problem is that similar local intensity patterns can occur at many places. One method to reduce the matching ambiguity is to use more than two images in order to establish correspondences [64, 60, 79]. Such methods are called *multi-baseline* stereo vision methods.

Stereo vision methods generally exploit the geometric relationship that exist between the 2-D projections of a 3-D point in different images to aid the correspondence search. Each point in an image defines a ray, which at some unknown distance intersects the object being imaged. The projection of such a ray onto the retinal surface of another camera constrains the locations where the imaged object may be found in the other image.

The search space for an image correspondence is therefore reduced from 2-D (the whole image) to 1-D (the projection of the ray into the image). This constraint is known as the *epipolar constraint* [32, 17]. Exploiting the epipolar constraint dramatically reduces the computational demands of the correspondence problem and diminishes the risk of establishing erroneous correspondences.

Most of the existing work on stereo vision is concerned with conventional perspective cameras. Conventional cameras have a narrow field of view, implying that in order to obtain a reconstruction of the surrounding environment many images are needed. When using panoramic images, fewer images are needed due to their 360° field of view. The use of panoramic images does however has some specific aspects. For conventional perspective cameras, the search for correspondences can be done along an *epipolar line* in the image. Panoramic cameras perform a non-linear projection which, generally, does not preserve straight lines. As a result, a ray projects to an *epipolar curve* rather than an epipolar line. An important issue in panoramic stereo vision is the parameterization of the epipolar curves in a way that permits efficient traversal in order to establish image correspondences.

In this chapter we present methods to obtain a 3-D scene reconstruction from two and more cylindrical panoramic images. In our application images are acquired by a single panoramic camera mounted on top of a mobile robot that moves around. The chapter is organized as follows. In section 4.2, we present a panoramic stereo vision algorithm for a pair of cylindrical panoramic images. We present a parameterization of the epipolar curve that can be used to guide the search for image correspondences and show how a 3-D reconstruction can subsequently be obtained using triangulation. In section 4.3, we present a multi-baseline stereo vision algorithm that uses more than two images. We derive a parameterization of the epipolar curve in terms of inverse depth. Using this parameterization the search for 2-D image correspondences across multiple images and the 3-D reconstruction can be performed efficiently. In our application, we use images acquired by a single moving camera. We therefore need a method to estimate the camera poses. A simple technique utilizing robot odometry and tracked image correspondences is presented in section 4.4. Experimental results obtained using our stereo methods are presented in section 4.5. A discussion and conclusions are presented in section 4.6.

4.2 Panoramic Stereo Vision

4.2.1 Epipolar geometry

Any pair of images obtained by a central projection are related by the *epipolar geometry* that depends only on the relative pose and internal parameters of the camera(s) by which the images were acquired. The epipolar geometry reduces the search space for image correspondences from 2-D to 1-D and can be formalized as follows.

Let C_0 and C_1 denote relative camera poses at which two images I_0 and I_1 are acquired by a central projection. Let $\mathbf{X}_i = [X, Y, Z]_i^T$ denote the coordinates of scene point X

expressed in the i-th ($i \in \{0,1\}$) camera pose and let $\mathbf{x}_i = [x,y,z]_i$ denote its projection onto the retinal surface. We designate C_0 as a reference camera pose, i.e. the coordinate system in which vectors are measured and we refer to I_0 as the *reference image.* Let $\mathbf{t}_1$ be the translation vector between C_0 and C_1 and let $\mathbf{R}_1$ be a rotation matrix that aligns the two coordinate frames. Then point X can be represented as

$$\mathbf{X}_0 = \mathbf{t}_1 + \mathbf{R}_1\mathbf{X}_1. \tag{4.1}$$

In practice, $\mathbf{X}_0$ and $\mathbf{X}_1$ are unknown. All that is available are their respective projections into the images $\mathbf{x}_0$ and $\mathbf{x}_1$. The projections are related to equation 4.1 by

$$r_0\mathbf{x}_0 = \mathbf{t}_1 + r_1\mathbf{R}_1\mathbf{x}_i, \tag{4.2}$$

where r_0 and r_1 are unknown scalars whose values are to be recovered by stereo triangulation.

The epipolar geometry expresses the fact that the rays spanned by $\mathbf{x}_0$ and $\mathbf{x}_1$ meet at the single point X in space. Put differently, $\mathbf{x}_0$, $\mathbf{x}_1$ and $\mathbf{t}_1$ are co-planar. The plane spanned by $\mathbf{x}_0$ and $\mathbf{t}_1$ and defined by its normal $\mathbf{n}_0 = \mathbf{t}_1 \times \mathbf{x}_0$ is called the *epipolar plane.* In the C_1 coordinate frame, the plane normal $\mathbf{n}_1 = [n_x, n_y, n_z]_1^T$ is derived as

$$\mathbf{n}_1 = \mathbf{R}_1^T(\mathbf{t}_1 \times \mathbf{x}_0). \tag{4.3}$$

The *epipolar constraint* is now established by the co-planarity condition

$$\mathbf{n}_1 \cdot \mathbf{x}_1 = 0. \tag{4.4}$$

Figure 4.1 illustrates that, for a conventional perspective camera, an *epipolar line* is formed by intersecting the *epipolar plane*, spanned by $\mathbf{x}_0$ and $\mathbf{t}_1$, with the retinal plane. Equivalently, the epipolar line is the image in one camera of a ray from the effective pinhole of the other camera and passing through an image point in the other camera. As the position of the 3-D point being projected varies, the epipolar plane appears to rotate about the baseline. The family of planes induced is called an *epipolar pencil.* Consequently, all epipolar lines meet at a single point called the *epipole.* The epipole is the point of intersection of the line joining the effective pinholes of the cameras — the translation vector $\mathbf{t}$, referred to as the *baseline* in stereo literature — with the retinal plane. Equivalently, the epipole is the image in one camera of the effective pinhole of the other camera. Notice that in the epipole direction depth cannot be estimated.

Figure 4.1 illustrates the epipolar geometry for a popular camera configuration involving two camera whose retinal planes are related by a horizontal displacement only. This configuration is known as the *parallel camera configuration.* In this configuration stereo matching is greatly simplified because all epipolar lines are parallel and horizontal. As a consequence, no explicit parameterization of epipolar lines is required; in order to establish a correspondence, matching can be performed along a single row of pixels and a depth measure can easily be derived from the disparity measured in pixels [17]. The resulting stereo algorithms typically achieve real time performance nowadays [45, 100, 57]. If two cameras are approximately in a parallel configuration, an exact parallel camera configuration can be achieved in software by re-projecting the images obtained from both cameras onto a common plane [17].

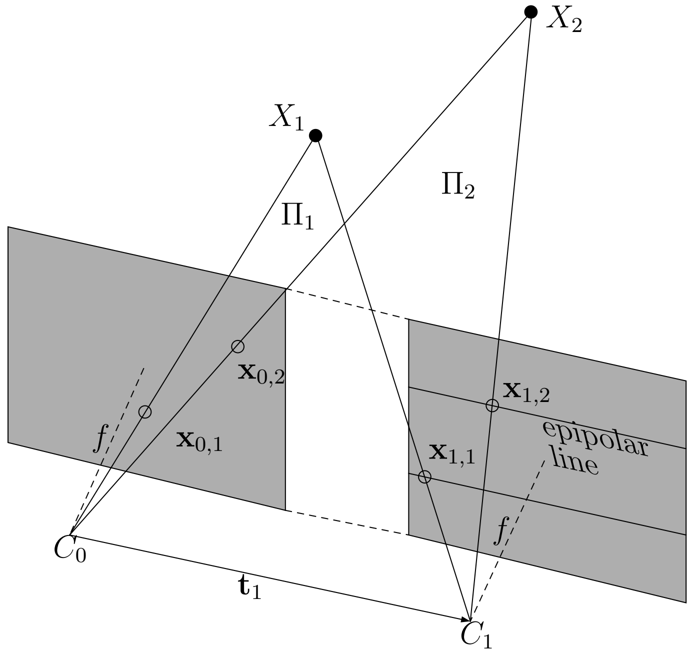


Figure 4.1: Epipolar geometry for a pair of cameras in a parallel configuration. The two retinal planes are displaced and co-planar in space. In this configuration, the epipole lies at infinity and all epipolar lines are parallel.

4.2.2 Epipolar curves

The geometric image formation performed by a panoramic camera generally does not preserve straight lines. Therefore, *epipolar curves* are obtained rather than epipolar lines. This raises the questions how these curves look like and how they can best be parameterized.

Svoboda [85] studies the epipolar geometry for a catadioptric vision sensor based on a hyperboloid mirror. The theory starts from the observation that the catadioptric vision sensor measures intensity from the effective viewpoint located inside the mirror, as discussed in chapter 2. Epipolar planes are therefore spanned by the baseline relating the effective viewpoints — the focus inside the mirror — and a point on the surface of the mirror. The epipolar plane intersects the mirror. Correspondences can thus be sought in the catadioptric image along the projection of the intersection into the image. It is shown that the resulting epipolar curves are general conics (ellipses, circles, hyperbolas and parabolas). In [82] a parameterization for each type of conic, and a selection mechanism to determine the type of conic, are proposed. Although not explicitly described, the proposed parameterizations can be used to sample points (search window centers) along the curve for the purpose of stereo matching.

Wei [106] proposes to re-project a catadioptric image onto a virtual paraboloid, then followed by an orthographic projection onto a virtual image plane. When using two such re-projected images obtained at different viewpoints, the epipolar curves become circles, albeit of different radii. The advantage of the re-projection step is that it obviates the need to determine the type of conic.

Ollis [65] presents two catadioptric stereo head camera configurations that yield a simple epipolar geometry. They describe a setup where two catadioptric cameras are stacked vertically such that the mirror axis of symmetry of the two mirrors coincide. They also describe a setup using two mirrors (a larger mirror above a smaller one) and a single camera. In these co-axis mirror configurations the epipolar curves reduce to straight lines emitting radially from the image center. In [29] a similar co-axis mirror pair with two cameras is used to obtain real-time panoramic stereo vision. They also perform a cylindrical re-projection of the catadioptric images, thus obtaining cylindrical panoramic images. In these cylindrical panoramic images, the radial epipolar lines are transformed to parallel vertical lines. The resulting stereo algorithms are very efficient because they do not require active correlation windows to perform stereo matching.

In our application we have a single catadioptric vision sensor mounted vertically on top of our mobile robot. Stereo vision thus has to be performed on images obtained at different robot poses. The robot operates in an indoor environment, hence its motion (and consequently that of the camera) is restricted to a horizontal displacement and rotation. Instead of using the catadioptric images for stereo matching directly, we propose to re-project the catadioptric images onto a virtual cylinder, thus obtaining virtual cylindrical panoramic images. Whereas a rotation of the robot causes a rotation in the catadioptric image domain, it causes a shift in the cylindrical panoramic image. The advantage of using the re-projection step is that stereo matching can be done without requiring rotating correlation windows (as proposed in e.g. [83]). The computational costs of performing a re-projection of the images is low compared to the costs of using rotating correlation windows for stereo matching. Moreover, off-the-shelf feature trackers (such as KLT [3]), originally developed for conventional perspective images, can be employed to obtain an initial set of image correspondences required to estimate the relative camera positions. A further advantage is that only a single type of epipolar curve (a sinusoid, as we will show in the next section) results.

4.2.3 Panoramic Stereo from a Pair of Images

Parameterization of the epipolar curve. In chapter 2 we defined a virtual cylindrical panoramic camera. The virtual camera is constructed by specifying a unit radius virtual cylinder in the mirror frame. Once a cylindrical panoramic image is created from the catadioptric image, we can proceed as if we truly have a cylindrical panoramic vision sensor instead of a catadioptric sensor. Let $\mathbf{X}_0$ and $\mathbf{X}_1$ denote the coordinates of a scene point X in the cylindrical camera coordinate frames. Let $\mathbf{x}_0$ and $\mathbf{x}_1$ denote the projection of X into images I_0 and I_1 respectively. Let $\mathbf{y}$ denote the cylindrical coordinate representation of the projection of X given by equation 2.42. Let $\mathbf{x}$ denote the Cartesian coordinates of $\mathbf{y}$, which can be derived using equation 2.43.

In the cylindrical panoramic image domain epipolar curves are sinusoids. This can be understood by expanding equation 4.4 and expressing $\mathbf{x}_1$ as $\mathbf{x}_1 = [\cos\phi_1, \sin\phi_1, z_1]$ giving

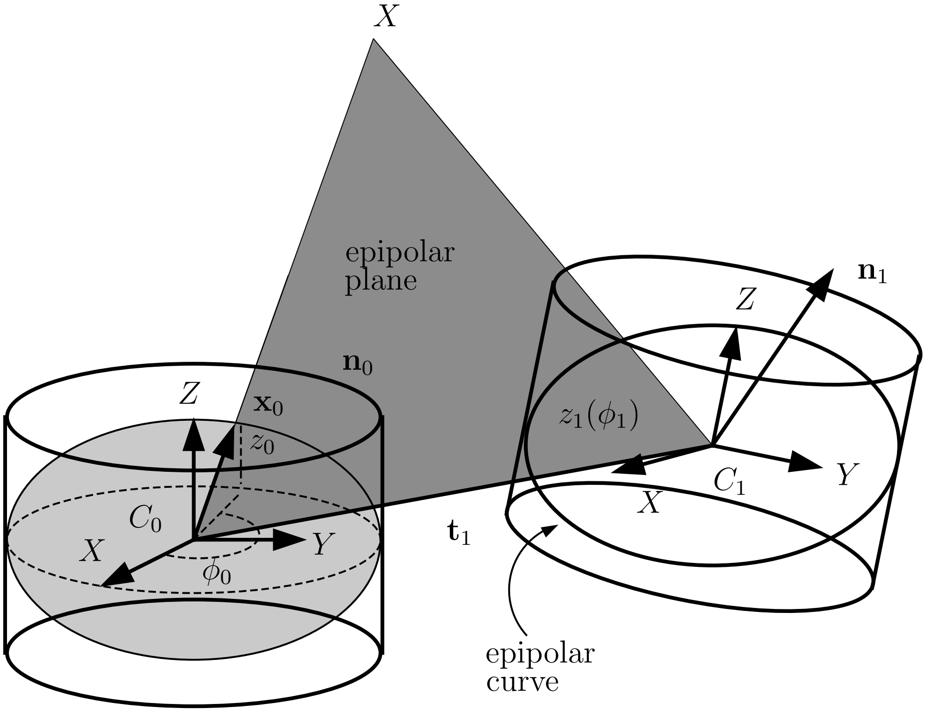


Figure 4.2 : Epipolar geometry for a pair of cylindrical panoramic images.

$$\begin{aligned} 0 &= \mathbf{n}_1 \cdot \mathbf{x}_1 = [n_x, n_y, n_z][\cos\phi_1, \sin\phi_1, z_1]^T \\ &= n_x \cos(\phi_1) + n_y \sin(\phi_1) + n_z z_1. \end{aligned} \tag{4.5}$$

The elevation z_1 can be expressed as a function of the angle ϕ_1 as

$$z_1(\phi_1) = -\frac{n_x \cos(\phi_1) + n_y \sin(\phi_1)}{n_z}. \tag{4.6}$$

Figure 4.2 displays the epipolar geometry for cylindrical panoramic images.

Using this parameterization a 1-D correspondence search along the epipolar curve can be performed. Using the known camera rotation and translation, a given image coordinate $\mathbf{y}_0 = [\phi_0, z_0]^T$ defines the epipolar plane normal $\mathbf{n}_1$. Candidate correspondences $\mathbf{y}_1 = [\phi_1, z(\phi_1)]^T$ living on the epipolar curve can then be generated using equation 4.6. The image similarity between local neighborhoods around $\mathbf{y}_0$ and each generated $\mathbf{y}_1$ can be used to select the best candidate. In the next section we review some common approaches and describe the matching method we adopt.

Establishing correspondences. Matching in stereo vision involves comparison of local image regions (or intrinsic local characteristics, i.e. , features) and determining which regions are most similar according to some matching criterion. Matching methods can be categorized as local or global. *Local matching methods* match each point in the reference image independently (see e.g. [47]). *Global matching methods* consider points at once. Each point in an image gives rise to an epipolar curve in the other image and vice versa. Global matching schemes consider all windows along such conjugate epipolar curves at once and seek a set of matches that optimizes some global constraints. Examples of such global constraints are uniqueness (there exists a one-to-one correspondence between points along conjugate epipolar curves), surface continuity (the disparity measure along conjugate epipolar curves is smooth) and ordering (features appear in the same order when traversing conjugate epipolar curves) [17, 57, 116]. These prior assumptions, although not always correct, can help to resolve local ambiguities. The drawback of global methods is that they are computationally more expensive than local methods because they consider a harder optimization problem.

A further distinction between matching methods is whether they compare image features or correlate small image windows. Image features, such as corners, are usually chosen to be insensitive to viewpoint and illumination changes. Due to the sparseness of salient features in images, *Feature based matching* methods generally produce sparse depth estimates. *Correlation based matching* methods produce dense depth estimates. Correlation based methods implicitly assume that the disparity is similar for each pixel in the window under consideration. This assumption is violated for large windows. Furthermore, using larger windows increases the computational cost. Small windows, on the other hand, have a lower signal-to-noise ratio and are more easily confused.

Our stereo algorithm uses correlation based matching, thus giving dense depth estimates. It aims to find a matching window for each pixel independently. Matching is done as follows. For a given point (ϕ_0, z_0) in the reference image, we first establish a search interval $[\phi_{1,d_{\min}}, \phi_{1,d_{\max}}]$, where $d_{\min}$ corresponds to a minimal distance of a scene point from C_0 and $d_{\max}$ corresponds to a scene point at some maximal distance. We thus limit the space of interest; for far away objects a reliable depth estimate cannot be obtained because of triangulation uncertainty, and (very) nearby objects are unexpected. A set of points is sampled along the epipolar curve $\{(\phi_i, z(\phi_i)) : \phi_i = \phi_{d_{\min}} + i\Delta\phi \in [\phi_{d_{\min}}, \phi_{d_{\max}}]\}$ where $\Delta\phi$ is a chosen angular resolution. The best match is found by computing the correlation values for each point and selecting the one that matches best. We choose the sum of absolute intensity differences (SAD) as a similarity criterion. Others, such as the sum of squared intensity differences (SSD) or normalized cross correlation could also have been used. The difference in performance is generally marginal [56].

Window based matching is a relatively expensive operation. In order to avoid unnecessary correlation computations, a correspondence search is only initiated when the part of the curve in the interval $[\phi_{i,d_{\min}}, \phi_{i,d_{\max}}]$ is contained entirely within the image domain (which is given by $-\pi \leq \phi < \pi, z_{\min} \leq z \leq z_{\max}$). This ensures that corresponding points, unless occluded by other objects, are guaranteed to be visible from both viewpoints so that a correct match can (in principle) be found. This visibility condition can be verified by first checking the if the end-points of the search interval $(\phi_{d_{\min}}, z(\phi_{d_{\min}}))$ and $(\phi_{d_{\max}}, z(\phi_{d_{\max}}))$ reside in the image domain. Next, we check if the epipolar curve has an extremum in the search interval and, if it does, whether the extremum falls within the image domain.

Obtaining a reconstruction. Suppose a corresponding pair of image coordinates $\mathbf{x}_0$ and $\mathbf{x}_1$ is available. The distances between C_0 and X, and C_1 and X (see figure 4.2) can be estimated from the angular difference under which point X is perceived from these poses. The cosine rule can be applied to compute the angle between the $\mathbf{x}_0$ and translation vector $\mathbf{t}_1$

$$\theta_0 = \arccos(\frac{\mathbf{x}_0 \cdot \mathbf{t}_1}{||\mathbf{x}_0||\,||\mathbf{t}_1||}). \tag{4.7}$$

Similarly, the angle between $\mathbf{x}_1$ and the rotated translation vector $\mathbf{R}_1\mathbf{t}_1$ is computed as

$$\theta_1 = \arccos(\frac{\mathbf{x}_1 \cdot \mathbf{R}_1\mathbf{t}_1}{||\mathbf{x}_1||\,||\mathbf{t}||}). \tag{4.8}$$

Finally, d_0 and d_1, corresponding to the distance to point X measured from poses C_0 and C_1 respectively, can be computed by application of the sine rule

$$d_0 \;=\; \tfrac{\sin\theta_1}{\sin(\theta_1-\theta_0)}||\mathbf{t}||, \tag{4.9}$$

$$d_1 \;=\; \tfrac{\sin\theta_0}{\sin(\theta_1-\theta_0)}||\mathbf{t}||. \tag{4.10}$$

The 3-D coordinates of point X can be reconstructed using the relationship

$$\mathbf{X}_i = d_i \frac{\mathbf{x}_i}{||\mathbf{x}_i||} \quad \text{where} \quad i \in \{0,1\}. \tag{4.11}$$

Note that distance d_i is related to depth r_i from equation 4.2 as

$$r_i = \frac{d_i}{||\mathbf{x}_i||} \quad \text{where} \quad i \in \{0,1\}. \tag{4.12}$$

Sensitivity and accuracy. Sensitivity with respect to errors in viewing angles and norm of the translation vector can be investigated by examining equation 4.9 (or 4.10).

Differentiating 4.9 with respect to with respect to $||\mathbf{t}||$ (known as the *baseline length* in stereo) gives

$$\frac{\partial d_0}{\partial ||\mathbf{t}||} = \frac{\sin\theta_1}{\sin(\theta_1-\theta_0)} = \frac{d_0}{||\mathbf{t}||}. \tag{4.13}$$

This shows that an error in $||\mathbf{t}||$ causes an error in the depth estimate that is proportional to the depth and inverse proportional to the baseline length.

Differentiating 4.9 with respect to θ_1 gives

$$\frac{\partial d_0}{\partial \theta_1} = d_0 \left(\frac{\cos\theta_1}{\sin\theta_1} - \frac{\cos(\theta_1-\theta_0)}{\sin(\theta_1-\theta_0)} \right). \tag{4.14}$$

This expression readily shows that if a scene point lies far away or in the direction of one of the epipoles, so that the difference between θ_0 and θ_1 is small, the resulting depth estimate is unreliable.

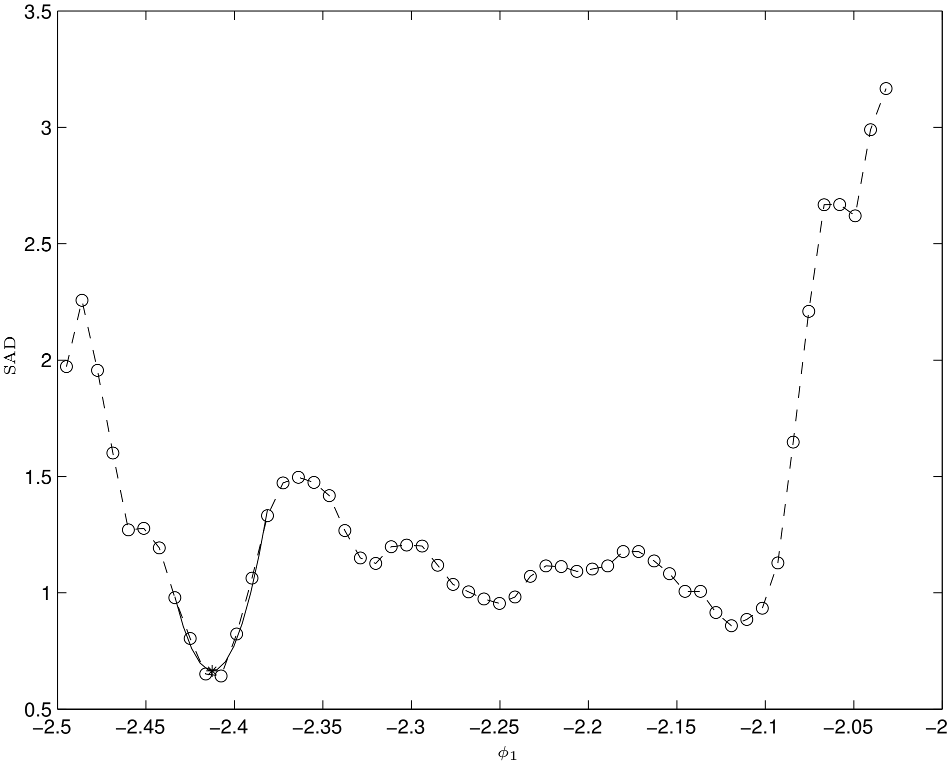


Figure 4.3: Sum of absolute difference (SAD) values obtained along the epipolar curve sampled at ϕ_1 and the fitted parabola.

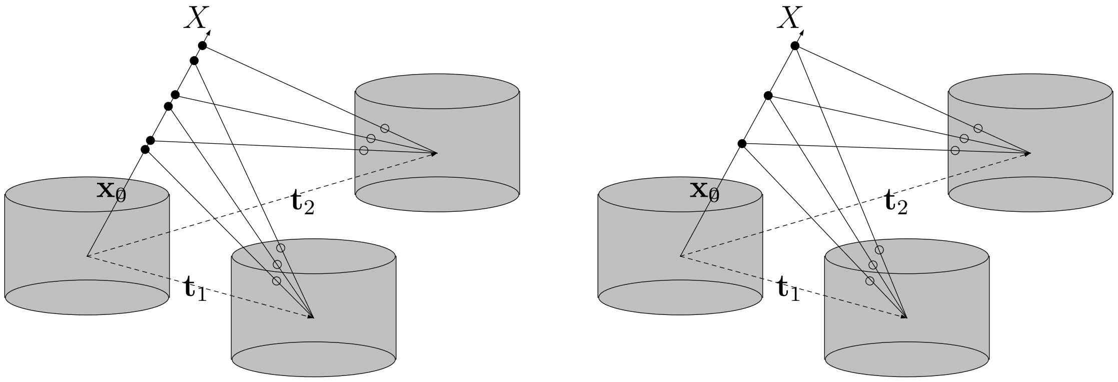
Combining equations 4.13 and 4.14, a first order approximation of the depth error Δd_0 is obtained as

$$\Delta d_0 = \frac{d_0}{\|\mathbf{t}\|}\Delta\|\mathbf{t}\| + d_0\left(\frac{\cos\theta_1}{\sin\theta_1} - \frac{\cos(\theta_1-\theta_0)}{\sin(\theta_1-\theta_0)}\right)\Delta\theta_1, \tag{4.15}$$

where $\Delta\|\mathbf{t}\|$ is the error in the baseline length and $\Delta\theta_1$ is the angular disparity error.

Consider the following typical situation. The camera has an exact translation of 0.5m so that $\|\mathbf{t}\| = 0.5$ and $\|\Delta\mathbf{t}\| = 0$. The distance to some object is exactly 3 m. An error of $\Delta\theta_1 = 1°$ in the estimation of the disparity then results in an depth error Δd_0 of about 0.3 m. The error can be reduced by extending the baseline. However, for a larger translation (baseline length) between viewpoints, matching is more difficult due to perspective deformation and occlusions. The above analysis shows that the disparity needs to be computed with high accuracy.

In finding the best match, there are two sources of error in the matching scheme we employ. First, correlation values are computed at nearest discrete pixel positions, not at the continuous point on the epipolar curve under investigation. A solution to this problem is to use some kind of interpolation. However, this would increase the computational demands considerably. A second error source is that the epipolar curve is only sampled at a finite number of points, i.e. the angular resolution is limited. One solution would be to



(a) angular

(b) inverse depth

Figure 4.4: a) The parameterization of epipolar curves as $z(\phi)$ is not a convenient one when more than two images are used. The problem is that the sampling that is implicitly performed along a ray from the reference viewpoint depends on the direction of the ray and the relative camera pose. As a result, it is not trivial to combine information obtained from different pairs of images. b) multi-baseline stereo overcomes this issue by projecting points along the ray to the images.

sample the epipolar curve more densely. However, without interpolation this would result in the same match value for multiple points along the epipolar line. With interpolation, denser sampling of the epipolar curve would increase the computational demands even further.

Our solution to these problems is to interpolate between match values along the epipolar curve by fitting a polynomial of degree 2 to the SAD values in a least-squares sense. In the fitting procedure, only neighboring points of the sample point that has the best match at discrete resolution contribute. By setting the derivatives of the polynomial with respect to ϕ_1 to zero, a new minimum for ϕ_1 can be computed. Via application of the epipolar curve equation this gives rise to a new point on the epipolar curve from which range can be estimated using the theory of the previous section. The computational overhead introduced by this refinement step yields a tiny fraction of the total computation time, which is dominated by the calculation of SAD values.

Figure 4.3 shows the correlation values as a function of ϕ_1 and a fitted parabola. Three neighbors around the minimum on both sides contribute to the estimated coefficients. Notice that the peakedness of the fitted parabola could be used to obtain an estimate of the (local) uncertainty of the estimated disparity [57].

4.3 Multi-baseline Panoramic Stereo Vision

The correspondence problem is a locally ambiguous problem because distinct regions in the scene can have a similar appearance in the images. Errors in the reconstruction of points obtained from stereo based on a single pair of images are therefore likely to be present. Furthermore, the reconstruction of points near the epipoles, which are always visible in the cylindrical panoramic images, is inaccurate due to large triangulation uncertainty. These limitations can be overcome by using more than two images to obtain a reconstruction. In this section, we present a new *multi-baseline panoramic stereo method* that reconstructs the environment from a set of K panoramic images.

One approach to incorporate more than two images is to apply the two view technique described in the previous section on the reference image I_0 and the $K-1$ other images. If each image pair provides a single depth estimate for each pixel in the reference image, combining the depth estimates is not trivial. There may be inconsistencies between the depth estimates resulting from different pairs. Furthermore the proposed parameterization of the epipolar curves used to govern the search for correspondences results in a different implicit sampling of points along the ray spanned by a $\mathbf{x}_0$ in the reference image for each alternative image. The problem is illustrated in figure 4.4(a).

Several methods have been proposed in literature to address these issues. The methods share in that they partition the 3-D space into discrete cells. For each cell a measure of consistency, reflecting the likelihood that the cell is occupied by an object, is then evaluated. We categorize these methods as 2D-3D, 3D-2D or 2D-2D methods.

2D-3D methods. 2D-3D methods start from image correspondences and use triangulation to reconstruct 3-D scene points. The scene space is partitioned into cells, and he number of points contained in a cell serves as evidence that the cell is occupied or empty. Triangulation uncertainty can be incorporated so that a reconstructed point does not only contribute evidence to the bin in which it is contained, but also to neighboring cells [60].

3D-2D methods. 3D-2D methods start from the 3-D cells and project the cells to the different images. Seitz and Dyer [79] formulate the scene reconstruction problem as a "voxel coloring" problem. Their method attempts to assign a unique color to each cell that is consistent with all input images. The underlying assumption in their approach is that when pixels are back-projected to the same bin, their color values should agree. A statistical measure of pixel color consistency is used to determine the occupancy state and color of the bin. By visiting the cells in depth order, occlusion is handled correctly. Cells can be ordered with respect to depth by assuming that no object lies within the convex hull of camera viewpoints. The method requires precise camera calibration and the accuracy and run-time are dependent on the cell resolution. Effects of quantization and calibration are most profound in regions with high spatial frequency. The "voxel coloring" algorithm reconstructs one particular scene consistent with the set of input images, namely the one closest to the camera convex hull is reconstructed.

2D-2D methods. 2D-2D methods exploit projective geometry in such a way that explicit 3-D reconstruction of points via triangulation or explicit projection of 3-D cells is

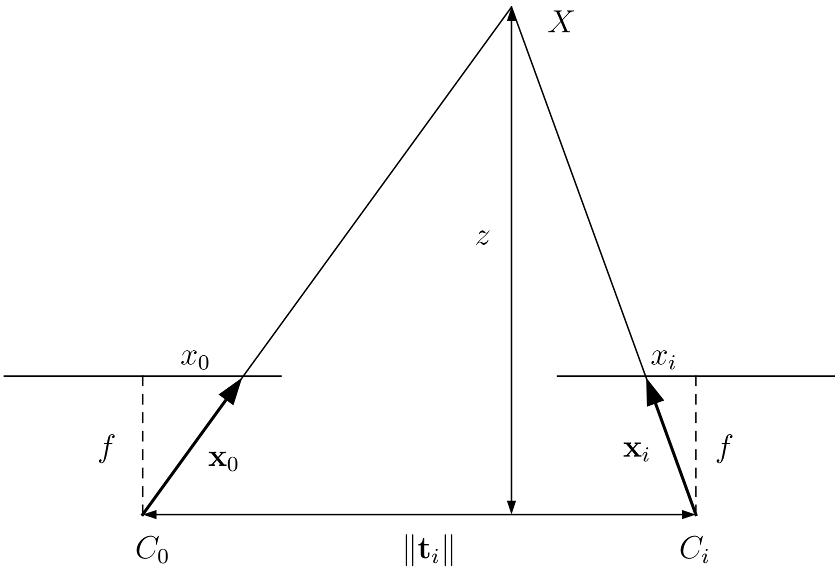


Figure 4.5: Disparity and depth measures for a parallel camera configuration. The depth measure z is defined as the distance of point X to the baseline. The disparity measure $\Delta x = x_i - x_0$ is related to z as $z = f \|\mathbf{t}_i\| / \Delta x$.

circumvented. An example is Collins' "space sweep" approach [8]. The method is based on the premise that areas of space where several image feature viewing rays (nearly) intersect are likely to be the 3-D locations of observed scene features. A single plane partitioned into cells is swept through the volume of space along a line perpendicular to a chosen plane. At each position of the plane along the sweeping path, the number of rays that intersect each cell are counted. The back-projection of point features from each image onto the sweeping plane is done efficiently by exploiting projective geometry relating central projections of points on planes (planar homography). A statistical measure is used to determine the likelihood that the cell is occupied by an object. The method proposed is restricted to image features such as corners and thus yields a sparse reconstruction.

Another example of a 2D-2D method is the multi-baseline stereo approach presented by Okutomi and Kanade [64]. The key benefits of their approach over 2D-3D and 3D-2D approaches are that the search for image correspondences across multiple images is performed efficiently and that depth estimates are obtained without explicit stereoscopic triangulation. The principle of multi-baseline stereo vision is illustrated in figure 4.4(b). In the next section we discuss their approach in more detail. Our approach can be regarded as an extension of their multi-baseline method to cylindrical panoramic images.

4.3.1 Okutomi and Kanade's multi-baseline stereo method

Okutomi and Kanade [64] present an efficient multi-baseline stereo method for multiple planar perspective cameras in a parallel camera configuration. The method exploits a projective invariant called the inverse depth. This scalar quantity expresses depth to a scene point as a fraction of the baseline length and is invariant under changes of the baseline length. As a result, depth estimates obtained using different baseline lengths can be combined.

Figure 4.5 sketches a top view of the parallel camera configuration. The disparity measure $\Delta x = x_i - x_0$ between the projection of a scene point X in the reference image (x_0, y_0, f) and the projection in the i-th image (x_i, y_i, f). The disparity measure Δx is related to the depth measure z to the scene point by

$$\Delta x = \|\mathbf{t}\| \, f \frac{1}{z}, \tag{4.16}$$

where $\|\mathbf{t}_i\|$ is the baseline length and f is the focal length. From equation 4.16 we see that the baseline length acts as a magnification factor in measuring disparity Δx in order to obtain z. The multi-baseline stereo method exploits the fact that if both sides of 4.16 are divided by $\|\mathbf{t}\|$ we obtain

$$\frac{\Delta x}{\|\mathbf{t}\|} = f \frac{1}{z} = \lambda, \tag{4.17}$$

where λ is a constant called the inverse depth. This means that for a particular point in the reference image, the disparity divided by the baseline length is constant because there is only one depth z for that point. Therefore, if search window similarity is represented with respect to λ, it should consistently yield good matching values at the correct value of λ independent of the baseline. Similarity measures obtained from different baselines can therefore be combined, and it can be expected that a unique match position results.

4.3.2 Panoramic multi-baseline stereo with different baseline directions

In this section we present a multi-baseline stereo vision method for cylindrical panoramic images. It is based on the multi-baseline stereo method summarized in the previous section. Our method extends the multi-baseline stereo method [64] in that we allow the baseline directions to vary. This is an essential prerequisite in order to cope with the uncertainty in depth estimates obtained in epipole directions. If all camera poses are co-linear (as in the parallel camera configuration), all images share the same epipoles so that reliable depth information cannot be obtained in the epipole directions, regardless of the number of images used. We derive a parameterization of the epipolar curve obtained in cylindrical panoramic images in terms of inverse depth. Using this parameterization, the search for image correspondences across multiple images can be performed efficiently and depth estimates are obtained without explicit stereoscopic triangulation.

The first step in our algorithm involves rotating virtual cylindrical panoramic cameras such that they all have the same orientation. After this rectification, the camera poses are related by translations only.

Using equations 2.43 and 4.1 the 3-D location of a point can be expressed as

$$r_0\mathcal{C}^{-1}(\mathbf{y}_0) - \mathbf{t}_i = r_i\mathcal{C}^{-1}(\mathbf{y}_i), \tag{4.18}$$

where $\mathbf{y}_0 = [\theta_0, z_0]^T$ and $\mathbf{y}_i = [\theta_i, z_i]^T$ are the cylindrical coordinates of points $\mathbf{x}_0$ and $\mathbf{x}_i$ on the surface of the unit radius cylinders centered at C_0 and C_i, and r_0 and r_i encode the depth to the projected scene point.

Dividing both sides of equation 4.18 by r_0 and applying the function $\mathcal{C}$ (equation 2.42) that transforms rays to cylinder coordinates to both sides gives

$$\mathcal{C}(\mathcal{C}^{-1}\mathbf{y}_0 - \frac{1}{r_0}\mathbf{t}_1) = \mathcal{C}(\frac{r_i}{r_0}\mathcal{C}^{-1}\mathbf{y}_i) = \mathbf{y}_i. \tag{4.19}$$

Note that $\mathcal{C}(\cdot)$ eliminates the quantity r_i/r_0 from the right hand side of equation 4.19. Equation 4.19 shows that the projection of a point on the cylindrical imaging surface at pose C_i is a function of $\mathbf{y}_0$, the translation vector $\mathbf{t}_i = [t_x, t_y, t_z]^T$ and the fraction $1/r_0$. The fraction $1/r$ is called the inverse depth, λ. The above equation can be expressed in vector form as

$$\begin{bmatrix} \arctan\left(\frac{\sin\phi_0 - \lambda t_y}{\cos\phi_0 - \lambda t_x}\right) \\ \frac{z_0 - \lambda t_z}{\sqrt{(\cos\phi_0 - \lambda t_x)^2 + (\sin\phi_0 - \lambda t_y)^2}} \end{bmatrix} = \begin{bmatrix} \phi_i \\ z_i \end{bmatrix} = \mathbf{y}_i. \tag{4.20}$$

Equation 4.20 parameterizes the sinusoidal epipolar curve in terms of inverse depth and can be used to govern the search for image correspondences across multiple images. Given an image coordinate $\mathbf{y}_0 = [\phi_0, z_0]^T$, the known translation vector $\mathbf{t}_i$ and a chosen value for λ, equation 4.20 yields the image coordinate $\mathbf{y}_i$ in the i-th image corresponding to a scene point at depth $1/\lambda$ from the reference pose.

In our multi-baseline stereo algorithm, for each pixel $\mathbf{y}_0$ from the reference image, equation 4.20 is used to generate a set of image coordinates $\mathbf{y}_i$ by plugging in multiple values for λ. Subsequently, image similarity is evaluated by computing the sum of squared differences (SSD) between windows centered at $\mathbf{y}_0$ and $\mathbf{y}_i$ respectively. The SSD values obtained from different images for the same $\mathbf{y}_0$ and λ are combined by adding them. The underlying assumption is that when an object is present at some depth $r = 1/\lambda$ from the reference pose, the window contents will have a roughly similar appearance in all images, consistently giving rise to small SSD values. Finally, the most likely inverse depth value $\lambda^\star$ for $\mathbf{y}_0$ is found by examining the summed SSD values obtained for each λ and selecting the one yielding the smallest value.

The complete multi-baseline stereo algorithm for K cylindrical panoramic images can be summarized as follows. Let $\mathcal{Y}$ denote the set of pixel coordinates in the reference image. Let Λ denote a set of inverse depth values whose values are determined to cover a minimal and maximal expected scene depth.

```
for all $\mathbf{y}_0 \in \mathcal{Y}$ do
  $\lambda^\star = \lambda_1$
  for all $\lambda \in \Lambda$ do
    $\mathcal{M}(\mathbf{y}_0, \lambda) = 0$
    for $i = 1$ to $K$ do
      compute $\mathbf{y}_i(\mathbf{y}_0, \mathbf{R}_i, \mathbf{t}_i)$ using equation 4.20
      $\mathcal{M}(\mathbf{y}_0, \lambda) \leftarrow \mathcal{M}(\mathbf{y}_0, \lambda) + \mathrm{SSD}(W(\mathbf{y}_0), W(\mathbf{y}_i))$
    end for
    if $\mathcal{M}(\mathbf{y}_0, \lambda) < \mathcal{M}(\mathbf{y}_0, \lambda^\star)$ then
      $\lambda^\star = \lambda$
    end if
  end for
end for
```

where $\mathrm{SSD}(W_0(\mathbf{y}_0), W_i(\mathbf{y}_i))$ computes the sum of squared differences between image windows W_0 and W_i centered at $\mathbf{y}_0$ and $\mathbf{y}_i$ respectively. The map $\mathcal{M}$ contains all summed SSD values. If desired, it is possible to perform some kind of regularization of $\mathcal{M}$.

4.4 Camera Pose Estimation

A prerequisite of (multi-baseline) stereo vision is that the camera poses at which images are acquired are known. Although our robot is equipped with fairly accurate odometry, the pose estimates provided by odometry are not accurate enough to fix the epipolar constraint. Due to small errors in the robot orientation estimates, epipolar curves may not pass through corresponding points. Therefore the relative camera poses need to be corrected. In this section we describe how we correct the relative camera pose estimates using corresponding image features.

4.4.1 Tracking features

An initial set of corresponding features is obtained by tracking salient image features through a sequence of images acquired by the robot during navigation. Tracking is performed using KLT [3], a C implementation of the feature tracker described by Shi and Tomasi [80], based on early work of Kanade and Lucas [54]. In their approach, salient image points are detected by examining the minimum eigenvalue of local intensity gradient matrices. Tracking is done using a Newton-Raphson method (see [73] for details), which minimizes the intensity discrepancy between an image window centered at a salient image point in the previous image and the current image. In spite of the multi-resolution approach implemented by KLT to allow for larger image displacements, the tracker fails when the overall image displacement is very large. Large displacements occur when the robot makes a sharp turn. Currently, we use wheel odometry measurements to counter-rotate the virtual cylindrical panoramic camera so that the overall image displacement is roughly compensated.

4.4.2 Estimation of the essential matrix

The relative camera pose relating an image I_k with the reference image I_0 can be recovered via their epipolar geometry. The epipolar geometry can be estimated from the set of T salient points that were successfully tracked between I_0 and I_k. Any true correspondence $\mathbf{x}_0 \leftrightarrow \mathbf{x}_k$ should satisfy the epipolar constraint from equation 4.4, which can be written as

$$\mathbf{x}_k^T \mathbf{R}(\mathbf{t} \times \mathbf{x}_0) = 0, \tag{4.21}$$

where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector relating the two camera poses. This can be expressed in matrix form as

$$\mathbf{x}_k^T \mathbf{RSx}_0 = \mathbf{x}_k^T \mathbf{Ex}_0 = 0, \tag{4.22}$$

where $\mathbf{S}$ denotes the (3×3) skew symmetric matrix for which $\mathbf{Sx}_0 = \mathbf{t} \times \mathbf{x}_0$. The matrix $\mathbf{E} = \mathbf{RS}$ is called the *essential matrix* [17].

8-point algorithm. In practice, correspondences $\mathbf{x}_{0,i} \leftrightarrow \mathbf{x}_{k,i}$ obtained by the feature tracker are noisy. As a result, each correspondence satisfies the co-planarity condition expressed by equation 4.22 only approximately. Let us rewrite equation 4.22 as

$$\mathbf{x}_{k,i}^T \mathbf{Ex}_{0,i} = \begin{bmatrix} x_k & y_k & z_k \end{bmatrix}_i \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}_i \approx 0. \tag{4.23}$$

Let us write the entries of the essential matrix as a vector $\mathbf{e} = [e_1, \ldots, e_9]^T$. Equation 4.22 can then be rewritten as

$$\begin{bmatrix} x_k x_0 & x_k y_0 & x_k z_0 & y_k x_0 & y_k y_0 & y_k z_0 & z_k x_0 & z_k y_0 & z_k z_0 \end{bmatrix}_i .\mathbf{e} = \mathbf{D}_i \mathbf{e} \approx 0. \tag{4.24}$$

Each correspondence thus provides one constraint on $\mathbf{E}$. Using T correspondences, equation 4.22 can be expressed as a linearly as $\mathbf{De} = 0$, where $\mathbf{D}$ is a $(T \times 9)$ design matrix constructed by stacking the T $\mathbf{D}_i$ vectors from equation 4.24.

A solution for $\mathbf{e}$, and thus to $\mathbf{E}$, can be found linearly by solving

$$\min_{\mathbf{e}} ||\mathbf{De}||^2 \quad \text{subject to} \quad ||\mathbf{e}|| = 1. \tag{4.25}$$

The constraint $||\mathbf{e}|| = 1$ is incorporated to fix the scale of $\mathbf{E}$ which removes one degree of freedom from $\mathbf{E}$ and a solution can be obtained when at least 8 correspondences are available. The minimum of $\mathbf{e}$ in equation 4.25 is the eigenvector of the moment matrix $\mathbf{M} = \mathbf{D}^T\mathbf{D}$ associated with the smallest eigenvalue and can be found by using a singular value decomposition (SVD) of $\mathbf{M}$. This algorithm is known as the 8-point algorithm [31].

An essential matrix has two equal eigenvalues and has rank two [17]. The linear 8-point algorithm does not enforce these properties on the recovered matrix $\mathbf{E}$. Let us express that the recovered matrix is not a true essential matrix by writing it as $\tilde{\mathbf{E}}$. The nearest true essential matrix, which we now express as $\mathbf{E}$, can be found via the singular value decomposition (SVD) of $\tilde{\mathbf{E}}$ as follows. Let the SVD decomposition of $\tilde{\mathbf{E}}$ be $\mathbf{U\Sigma V}^T$, where $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$. The nearest true essential matrix $\mathbf{E}$ can the be found as $\mathbf{E} = \mathbf{U\Sigma}'\mathbf{V}^T$, where $\mathbf{\Sigma}' = \text{diag}((\sigma_1 + \sigma_2)/2, (\sigma_1 + \sigma_2)/2, 0)$.

Robust estimation. The 8-point algorithm presented in the previous paragraph is very sensitive to noise in the image coordinates. In [31] a point normalization method is proposed for homogeneous image coordinates which is shown to decrease the sensitivity. We cannot apply the normalization scheme because we do not have homogeneous image coordinates but “real” 3-D vectors whose tip touches the cylindrical projection surface. As a normalization step, we normalize the vectors by dividing them by their length, which has been shown to decrease the noise sensitivity [68].

In the set of tracked points, there are bound to be erroneous correspondences (outliers). These outliers can be identified and discarded using robust estimation techniques. We use the Least of Median Squares (LMedS) method [78] to obtain a initial estimate of the essential matrix that is compatible with the majority of correspondences. The correspondences compatible with the initial estimate are subsequently used by an M-estimator [94, 114]. The M-estimator is implemented as an iteratively re-weighted variant of the 8-point algorithm. It identifies the small fraction of outliers that may have survived the LMedS step and assigns them low weights so that they their contribution to the final estimate of the essential matrix is reduced.

4.4.3 Recovering the camera motion

The rotation matrix $\mathbf{R}$ and $\mathbf{S}$ matrix can be determined from the singular value decomposition $\mathbf{E} = \mathbf{U}\mathbf{\Sigma}'\mathbf{V}^T$ as follows [30].

$$\mathbf{R} = \mathbf{U}\mathbf{Y}\mathbf{V}^T \quad \text{or} \quad \mathbf{U}\mathbf{Y}^T\mathbf{V}^T \tag{4.26}$$

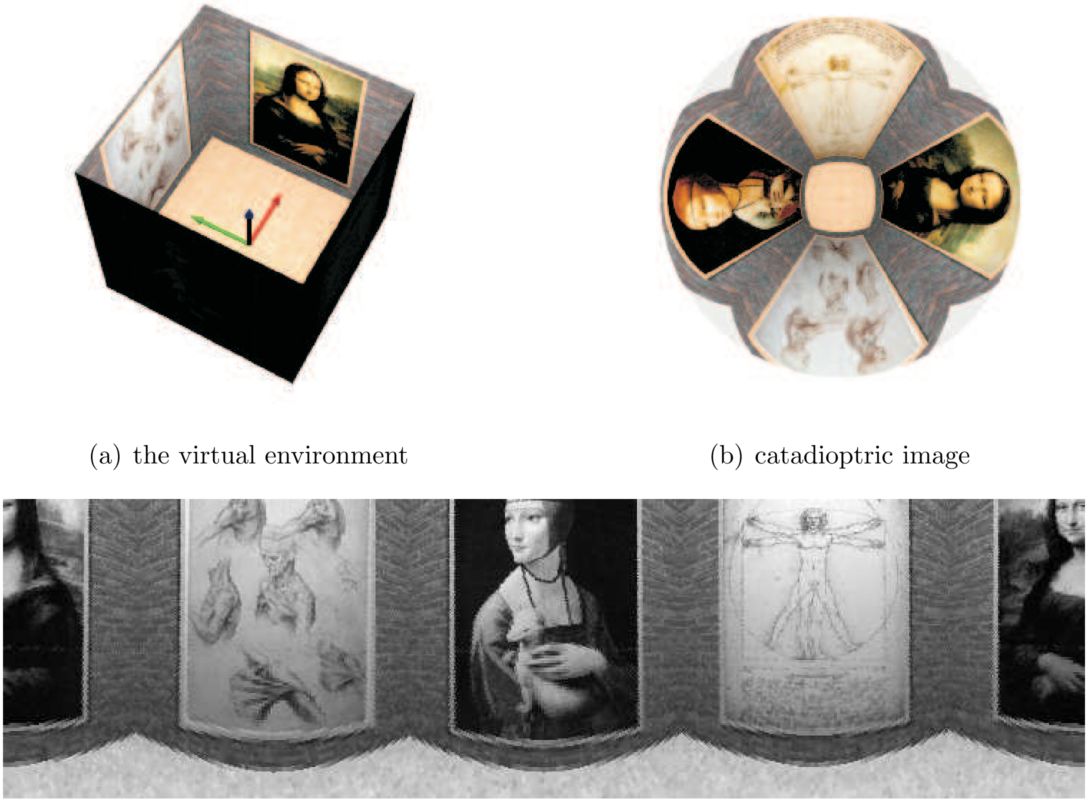
$$\mathbf{S} = \mathbf{V}\mathbf{Z}\mathbf{V}^T \quad \text{or} \quad \mathbf{V}\mathbf{Z}^T\mathbf{V}^T, \tag{4.27}$$

where

$$\mathbf{Y} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.28}$$

There are thus four possible pairings of $\mathbf{R}$ and $\mathbf{S}$ matrices that are compatible with the essential matrix. In order to select the correct pair, i.e. the actual rotation and translation, the geometric interpretation of each pairing must be investigated. This can be done by computing the depth to a tracked point in two images according to the relative poses implied by each pair using (for instance) the triangulation method presented in section 4.2.3. For noise free image correspondences, the correct pair should give a positive depth from both poses for any selected point. In practice, correspondences are noisy. We therefore compute the depth for all correspondences and adopt a majority scheme that selects the rotation and translation pair yielding the most positive depths from both camera viewpoints.

The estimated essential matrix is only defined up to an arbitrary scale factor. As a consequence, the length of the translation vector relating two images cannot be determined from an estimated essential matrix. Currently, we use wheel odometry to provide the length of the translation vector.



(a) the virtual environment

(b) catadioptric image

(c) cylindrical panoramic image

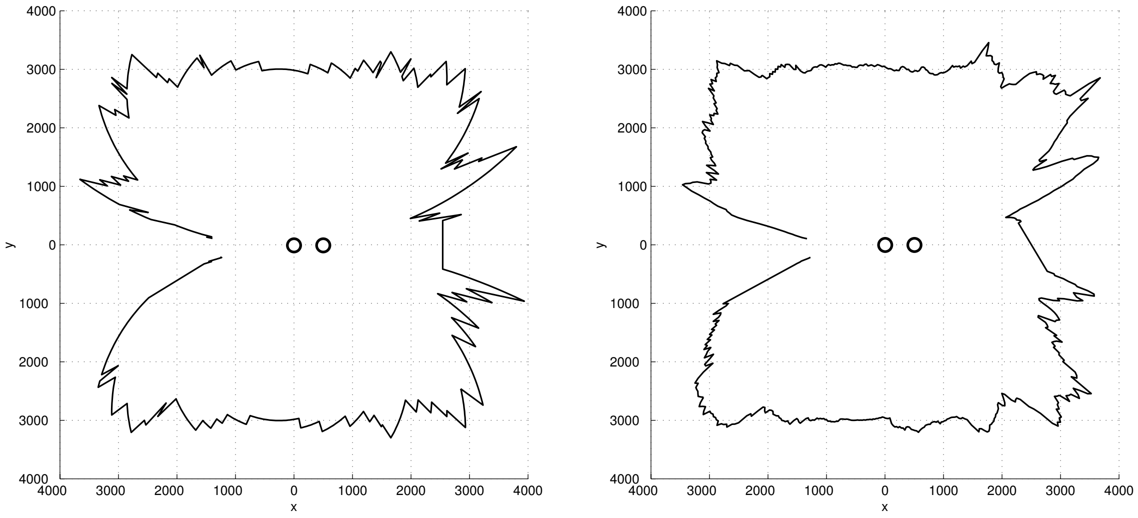
Figure 4.6: The virtual environment and various representations of an image acquired in the environment.

4.5 Experiments

4.5.1 Stereo experiment

We have tested the stereo technique presented in section 4.2 on synthetic, but realistic, image data. The POV-Ray ray tracer [72] was used to render images obtained by an catadioptric vision sensor in a virtual environment. The virtual environment has the shape of a cube ($6000 \times 6000 \times 6000$ in size) and has textured sides and bottom. Figure 4.6 displays an overview of the virtual environment and the different representations of an image acquired in the environment.

Depth is estimated from the cylindrical panoramic images (720×200 pixels) derived from catadioptric images (640×480 pixels) acquired at locations $[0, 0, 1000]$ and $[500, 0, 1000]$. Figure 4.7 shows the depth estimates obtained for a single row of pixels taken from the cylindrical panoramic image obtained at $[0, 0, 1000]$ (results for other rows give similar depth estimates). Figure 4.7(a) shows the range profile estimated at pixel resolution. Depth in directions perpendicular to the camera motion is estimated reliable with good accuracy. In the motion direction, the depth estimates are unreliable due to triangulation



(a) Depth estimated at pixel resolution (b) Depth estimated at sub-pixel resolution

Figure 4.7: Depth estimation results for a single scan-line. (a) without interpolation, (b) with interpolation. The reference pose (0,0) and the other camera pose (0,500) are marked by ∘ symbols.

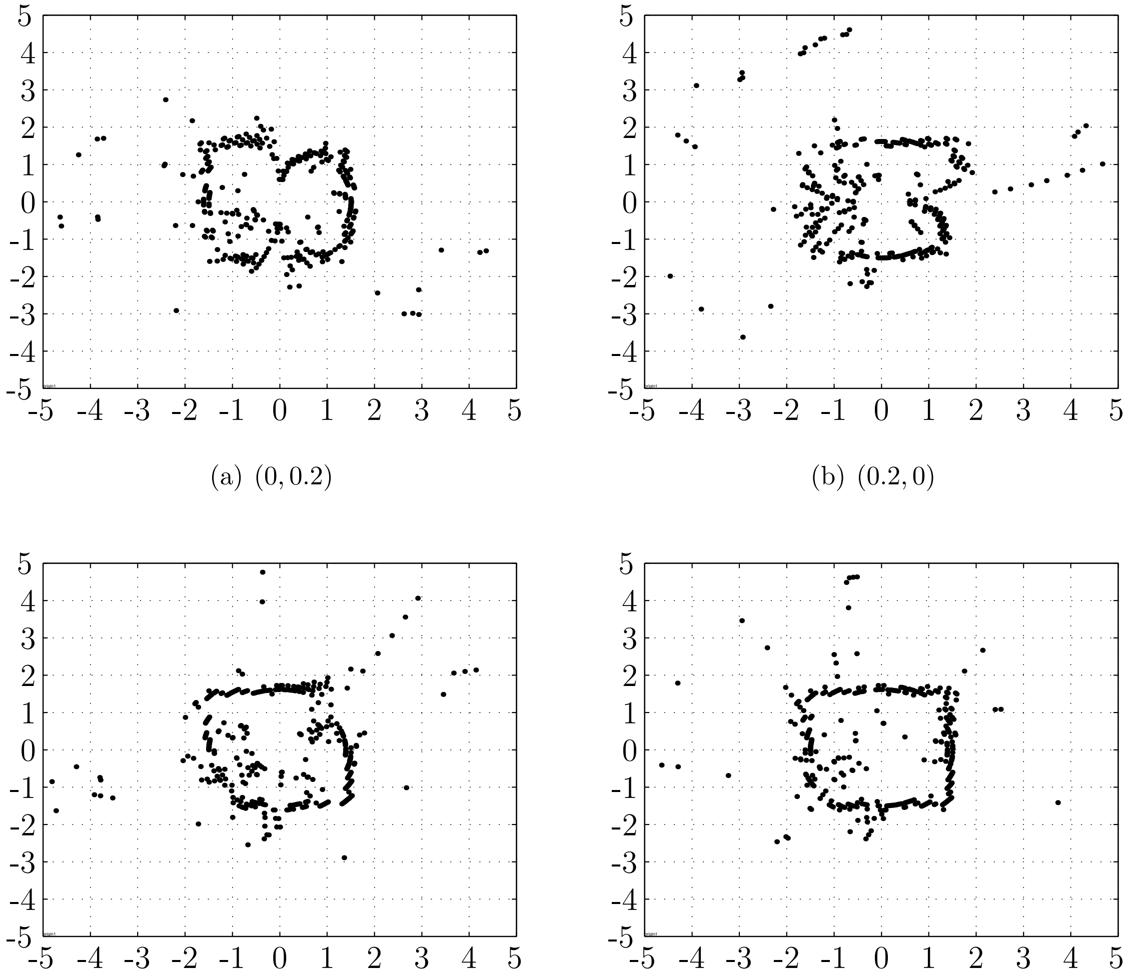
uncertainty. We have verified that the discontinuities in the range profile are caused by discretization. Figure 4.7(b) shows the improved estimate obtained using our sub-pixel method.

4.5.2 Multi-Baseline stereo experiments

We applied our multi-baseline stereo method both on rendered images and on real images acquired during robot navigation.

Simulation experiments. For the simulation experiments, we used the same environment as described in section 4.5.1, but scaled the cube to 3 × 3 × 3 units. Depth is computed from the cylindrical panoramic images (720 × 200 pixels) derived from catadioptric images (600 × 450 pixels) acquired at locations $[0, 0, 0.8]$, $[0, 0.2, 0.8]$, $[0.2, 0, 0.8]$ and $[0.2, 0.2, 0.8]$ (all under the same orientation). The image obtained at pose $[0, 0, 0]$ is used as a reference view.

The results of our multi-baseline stereo method are shown in figures 4.8(a)–4.8(d). The figures present a top-view of depth estimates obtained for viewing directions parallel to the floor (results for other rows give similar depth estimates). For visualization, for every pixel in the reference image, only the point for which the sum of SSD values found is minimal is displayed. Note that a depth estimate is provided for every viewing direction; no reliability measure has been used to reject unreliable depth estimates. Also notice the noisy depth estimates in the direction of the epipoles in figures 4.8(a)–4.8(c). The map



(a) $(0, 0.2)$

(b) $(0.2, 0)$

(c) $(0.2, 0.2)$

(d) combined

Figure 4.8 : Reconstruction of a virtual environment using our multi-baseline approach.

obtained after combining the maps is shown in figure 4.8(d). The resulting map is clearly better than any of the individual maps.

Reconstruction of a hallway. We tested our method at the end of a hallway in our building. A layout of the hallway is shown in figure 4.9. The positions where the images were acquired are indicated by circles. This simple environment lacks large depth discontinuities and occluding objects. Five cylindrical panoramic images were acquired by the robot. The catadioptric images (600×450 pixels) obtained by the vision sensor are transformed into cylindrical panoramic images (720×120 pixels). The motion relating the images to the reference one was estimated using the method presented in section 4.4.

A set of 25 inverse depth values was used in the stereo search. The set is obtained via an exponential sampling of depths r in the range 0.5m–20m. By performing exponential

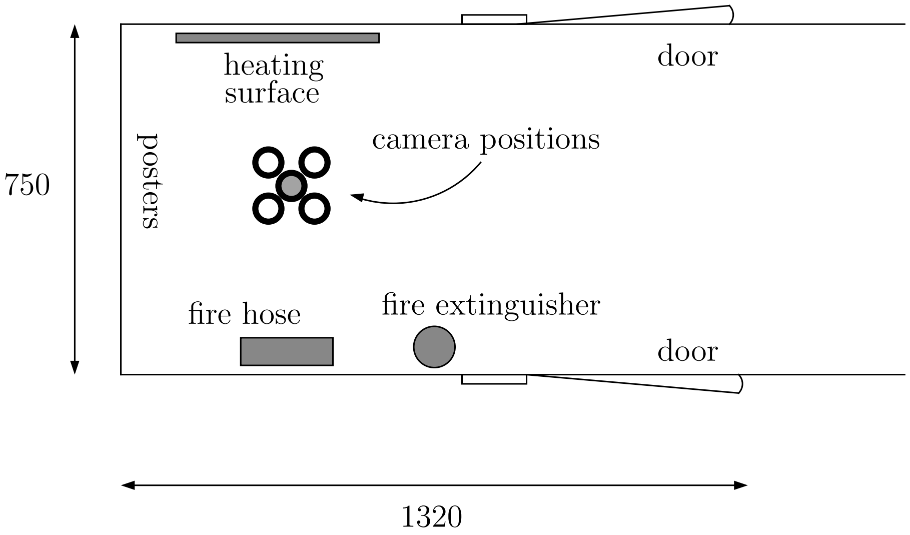


Figure 4.9: Layout of the hallway. Five image were acquired at the positions marked by the small circles. The central (arced) circle denotes the location of the reference image.

sampling, the resulting sampling of points along an epipolar curve is almost uniformly spaced. In the experiment we used 11 × 11 pixel correlation windows centered at the resulting nearest pixel coordinates to evaluate the sum of squared differences between image windows. Using this correlation window size gave a good tradeoff between accuracy and smoothness of the resulting depth maps.

Figure 4.10 displays the depth map computed from the reference image and the 4 other images. In the map, nearby objects appear brighter than objects further away from the reference pose. The depth map evidently shows that the overall geometry of the hallway is captured well. The far end of the hallway can be recognized near the sides of the depth map. The corners of the hallway at the near end appear slightly darker than the sections of the walls closer to the reference pose. The heating surface is not reconstructed well due to its repeating texture. Furthermore, specular reflections on the posters are not handled well; they appear as large depth discontinuities in the depth map. Due to the lack of texture, only some of the structure of the fire hose is visible in the combined depth map.

reconstruction of a laboratory. A similar experiment was performed using images acquired in a laboratory with large depth discontinuities and occlusions. A set of 22 images was acquired while the robot traversed a circular trajectory with radius of 1.3m. The first image was designated as the reference image. Figure 4.11 shows the reference image. A rectified image, obtained at pose (−1.21m,1.70m,102.0°) with respect to the reference pose, is shown in figure 4.11. The depth map computed from these images is shown in figure 4.12. As can be observed, the depth map contains many errors (arising due to occlusions, lack of texture, repeating texture, specular reflections etc). Such noisy depth maps are typically obtained from cylindrical panoramic image pairs. Application

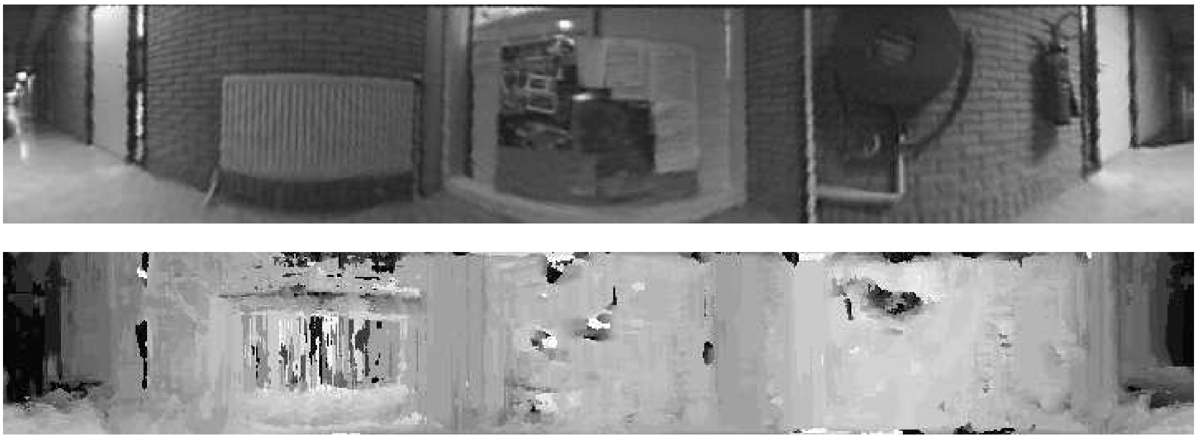


Figure 4.10: (top) The reference image. (bottom) The depth map computed from the reference image and 4 other images.

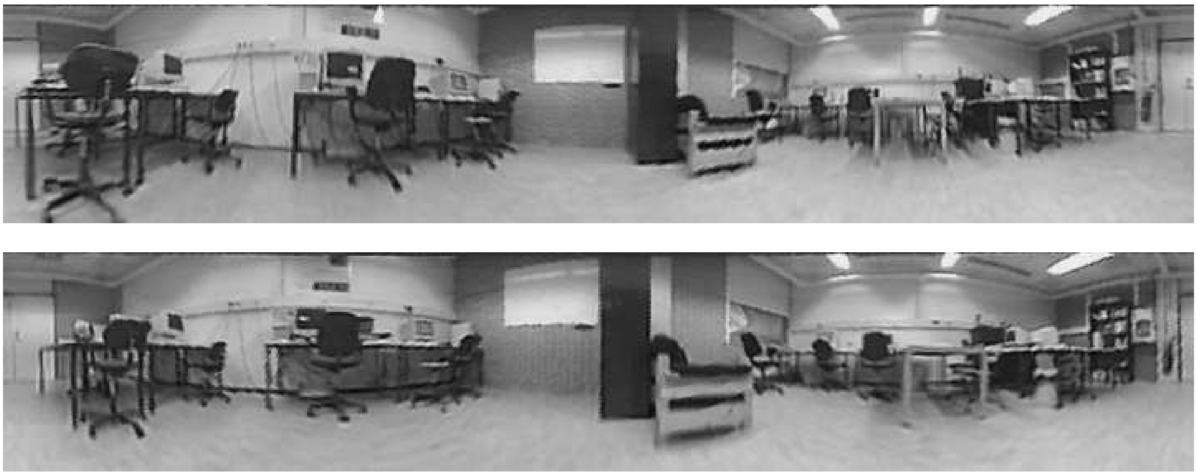


Figure 4.11: (top) The reference image. (bottom) A rectified image acquired at relative pose $(-1.21\text{m}, 1.70\text{m}, 102.0^\circ)$ from the reference pose.

of our multi-baseline stereo technique improves the estimated depth maps. In figure 4.12, the depth map computed from the reference image and all other image is shown. Overall, the geometry of the environment is captured well but some gross errors remain. These errors occur mainly at large depth discontinuities, due to occlusions (such as the chair on the left side in the images), and due to lack of texture (such as on the floor).

Equation 4.20 can also be used to warp the reference image to a target image that would be obtained at a nearby target position. We adopt the method presented in [58]. The mapping from the reference image to the target image is potentially one-to-many and visibility has to be considered. The principle of the warping method is similar in spirit to the painters algorithm [20] and handles occlusion without resorting to explicit 3-D computations. Briefly, a natural back to front ordering can be established. The warping procedure exploits the ordering by ensuring that far away objects are warped

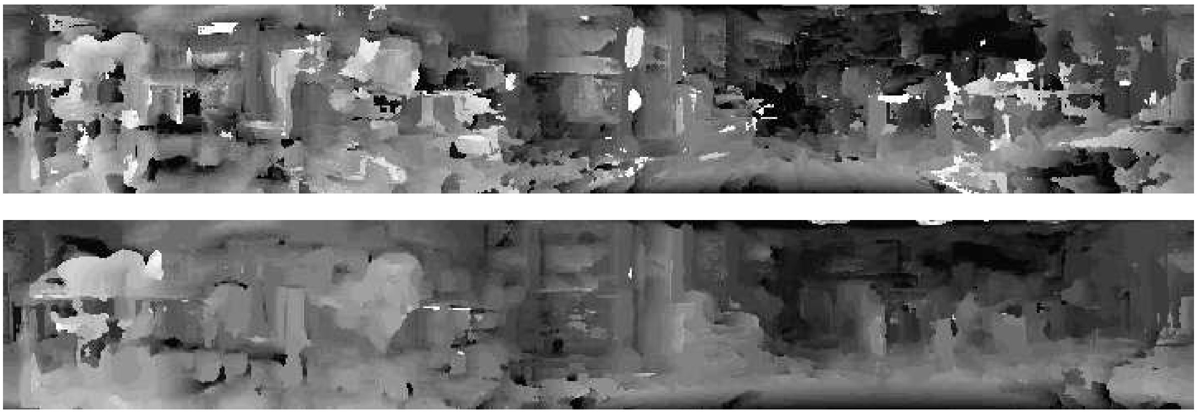


Figure 4.12: (top) The depth map estimated from the reference image and the rectified image shown in figure 4.11. (bottom) The depth map estimated from the reference image and 21 other images.

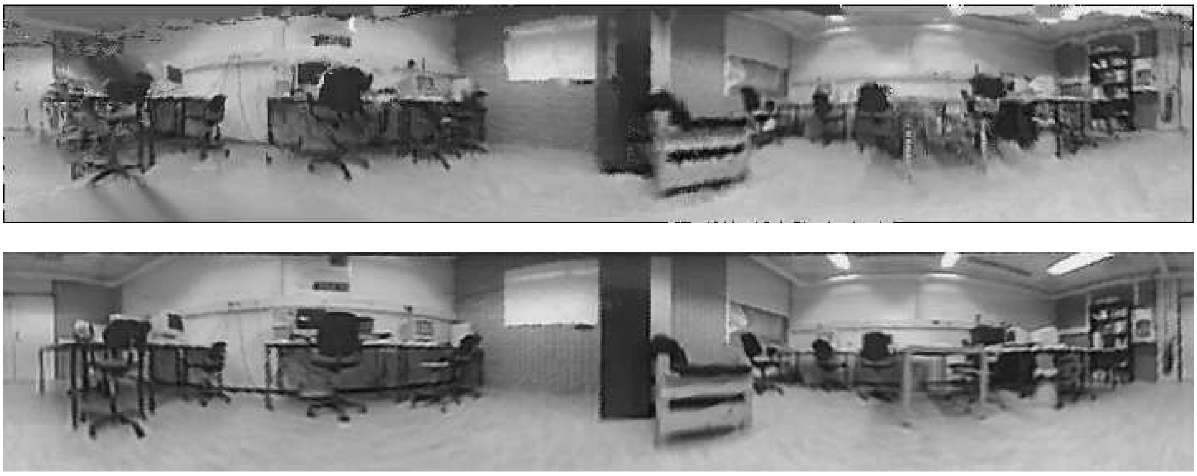


Figure 4.13: (top) Image obtained by warping the reference image to the pose associated with a target image (both images are shown in figure 4.11) using a depth map. The depth map was estimated from 21 image pairs and is shown in figure 4.12. (bottom) The target image from figure 4.11 duplicated for comparison.

first and allowing nearer objects to be "painted" over the further objects already drawn. In figure 4.13 we show the reference image warped to the pose associated with time image shown in figure 4.11. The depth map used in the warping was estimated from all 21 image pairs. We used a forward mapping scheme to warp the reference image. The forward mapping leaves holes in the warped image. In order the render a visually more appealing image, the holes were filled by interpolating gray values from neighboring pixels. If we compare the images shown in figures 4.11 and 4.13, we see that the warped image gives a reasonable prediction of the appearance of the scene as it would be observed from the target pose.

4.6 Discussion and Conclusions

In this chapter we have presented methods to obtain a 3-D scene reconstruction from cylindrical panoramic images obtained by a single moving catadioptric vision sensor. We have analyzed the epipolar geometry for cylindrical panoramic images and have shown that it gives rise to sinusoidal epipolar curves. We have presented two stereo vision algorithms that use a different parameterization of these epipolar curves. The first algorithm uses an angular parameterization and is suitable to obtain a 3-D reconstruction from a pair of images. We have shown how the accuracy of depth estimates can be improved with little computational overhead using a sub-pixel technique. A depth map estimated from a pair of images is noisy. Moreover, reliable depth estimates cannot be obtained in the direction of motion. The second algorithm addresses these issues by using more than two images to obtain a reconstruction. Our approach is based on the multi-baseline stereo method [64] for planar perspective cameras in a parallel camera configuration. We generalize their multi-baseline stereo method so that cylindrical panoramic images with different baseline directions to be used. The core of our algorithm is a novel parameterization of epipolar curves in terms of inverse depth. Using this parameterization the search for 2-D image correspondences and the 3-D reconstruction from multiple images can be performed efficiently.

Depth maps estimated by our multi-baseline stereo method are less noisy than those estimated from a just a pair of images for several reasons. Finding the correct match for a window containing an intensity pattern that is encountered many times along an epipolar curve in another image is difficult. Multi-baseline stereo often implicitly handles this situation because it selects the depth for which good matching value are found in all images. In practice, good matching values are obtained for depths corresponding to actual objects in the scene. The approach cannot handle homogeneous intensity patterns properly though. It is also impossible to obtain reliable depth estimates in the camera motion direction. Scene points along this direction all project to a tiny section of the epipolar curve. As a result, the image similarity values for points along the section of the epipolar curve will all be roughly similar. Our multi-baseline method implicitly handles this issue by using different baseline directions. The improvement that can be achieved using multi-baseline stereo with respect to stereo from a single pair of images has been demonstrated by experimental results.

In our current implementation, a fixed set of 25 inverse depths is used to compute depth maps. Because all image similarity values for all depths are maintained, it should be possible to improve the final depth map by regularization. The current (un-optimized) implementation of the multi-baseline method required about 20 seconds on a Pentium II at 233 MHz per depth map. In order to achieve real-time performance, a multi-scale approach involving an image pyramid could be used. Matches found at a course resolution level can then be refined at a higher resolution level. It would also be interesting to extend the method in such a way that for each pixel in the reference image, only a small set of inverse depth values, likely to correspond to an object, is maintained. Using gradient information, refinements of inverse depth can be obtained by the method of differences [54]. Such an approach resembles conditional density propagation by particle filtering, which has been successfully applied in the field of visual object tracking [37] and mobile robot localization [22, 105].

We have demonstrated that the estimated depth map can be used to predict the appearance of an image that would be observed at a pose close to the reference pose using image-based warping. An interesting application of this technique would be to use it in the context of appearance modeling for robot localization. A drawback of appearance-based modeling is that many training images are needed to estimate the model. In [10], many synthetic range profiles are generated by warping a few measured range profiles. In a similar manner, image-based warping could be used to generate many synthetic *images* from a few measured images and the estimated depth maps.

Chapter 5

Visual Odometry from a Catadioptric Vision Sensor

5.1 Introduction

In previous chapters we have presented methods to learn an appearance model for robot localization (chapter 3) and to obtain a 3-D scene reconstruction (chapter 4). Both methods require a collection of camera images obtained at known poses. In this chapter we describe a method to estimate the camera poses automatically from a sequence of images acquired during robot navigation in an initially unknown environment, without requiring wheel odometry or manual measurement of the poses.

Several techniques have been described in literature that aim to recover both the camera poses and the structure of the scene from corresponding features across a set of images (see [32] for an overview). This (structure-from-motion) problem is intrinsically difficult. In order to estimate the structure of the scene, the camera poses have to be known. Contrary, in order to estimate the camera poses, the structure of the scene has to be known. Techniques proposed in literature therefore typically interleave estimation of the scene structure and estimation of the camera poses. They involve solving a nonlinear optimization problem in many parameters, which is computationally expensive. Furthermore, in order not to get stuck in a local minimum they require a reasonable initial estimate of both the camera poses and the scene structure. From a practical point of view, the main difficulty in applying these techniques lies in establishing a reliable set of feature correspondences across the images.

One option to obtain a reliable set of correspondences is by manually registering many features across the images, but this is a tedious process. Another option is to try and establish correspondences automatically using correlation techniques. Such unguided matching may yield many erroneous correspondences. If images arrive sequentially, a third option is to track salient features through the image sequence. Feature tracking can be based solely on image correlation techniques. It can be made more robust if an initial estimate of the relative camera poses and the scene structure are available because these would constrain the positions of tracked features in a subsequent frame. Then the features tracked into subsequent images can be used to expand and refine the model containing the 3-D positions of the features and the past camera poses.

A drawback of such a sequential approach is that errors accumulate so that the model may eventually become inconsistent. One particularly prominent issue if a single moving camera is used, is that the *scale* of the reconstruction and camera pose estimates are subject to drift. The reason is that without knowledge about the lengths of the baselines that relate pairs of images, the reconstruction and camera pose estimates are subject to an arbitrary scaling.

In this chapter we present a method to estimate the rotation and translation between two subsequent camera poses from a pair of images. Our method assumes that the height of the camera with respect to a plane, i.e. the ground plane, is fixed. This situation is typical for a camera mounted on top of a moving mobile robot. By virtue of the fixed height assumption, the scale of baseline between the two camera poses can be related to the constant (albeit unknown) height. Estimating relative camera poses then boils down to finding an estimate of the parameters of a model that registers projections of points coming from the ground plane in the two images.

We propose a two-stage procedure to estimate the relative camera poses that takes advantage of the 360° field of view offered by a panoramic catadioptric vision system. The first stage estimates the relative rotation and baseline direction from cylindrical panoramic images, that are derived from the catadioptric images. Unguided tracking of image features between such panoramic images is relatively easy and robust. Furthermore, it has been shown in literature that a robust estimate of the rotation and direction of translation can be achieved from omnidirectional vision [19]. Once the relative rotation and baseline direction are known, only the scale of the baseline remains to be estimated. The second stage of our algorithm determines the baseline scale by registering planar perspective images of the ground plane, that are also derived from the catadioptric images.

This chapter is organized as follows. Section 5.2 reviews the geometric relationship between 3-D scene points and their projections in two images. In section 5.3 we present our two-stage approach. The first stage estimates the relative pose relationship from corresponding points in two cylindrical panoramic images via the epipolar geometry. The second stage estimates the baseline scale via the homography between planar perspective images of the ground plane. Experiments are presented in section 5.4. A simulation experiments provides insight in the sensitivity of the estimated baseline scale (stage two) with in the presence of objects extruding from the ground plane and specular reflections. We also apply our method on images acquired by our robot to obtain *visual odometry.* Unlike wheel odometry, visual odometry can be corrected upon recognizing a previously visited place. Conclusions and a discussion are presented in section 5.5.

5.2 Relative Camera Pose Estimation from a Pair of Images

This section aims to give the reader insight in the geometric relationship that exists between corresponding projections of 3-D scene points in two images. The projection of a scene point into an image depends on the type of camera used, its intrinsic parameters and the pose of the camera. We consider the situation where a single camera is positioned at two different poses. We assume that the camera performs a central projection (such as the cameras described in chapter 2) and that it is calibrated. The camera has an associated camera coordinate frame whose origin coincides with its center of projection. A point on the retinal surface of the camera can thus be characterized by a 3-D Euclidean vector $\mathbf{x}$ expressed in the camera coordinate frame and a ray in Euclidean space is characterized by $r\mathbf{x}(r > 0)$.

Let C and C' denote the Cartesian coordinate frames associated with two camera poses. The image obtained at C is denoted by I. Let $\mathbf{X}$ denote the 3-D Cartesian representation of scene point X expressed in coordinate frame C. The projection of X onto the retinal surface of the camera at pose C is denoted as $\mathbf{x}$. For some particular value of r, which encodes depth, it holds that $r\mathbf{x} = \mathbf{X}$. Similarly, $\mathbf{X}'$ denotes the 3-D Cartesian representation of X expressed in the C' coordinate frame, $\mathbf{x}'$ denotes its projection, and r' denotes the depth. The Euclidean transformation between the two coordinate frames is given by

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{t}, \tag{5.1}$$

where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector.

Let $\mathbf{b} = \mathbf{t}/b$, where $b = \|\mathbf{t}\|$, denote the unit length baseline relating the two coordinate frames. We refer to b as the *baseline scale*. In terms of the projections $\mathbf{x}$ and $\mathbf{x}'$ of X, equation 5.1 can be expressed as

$$r'\mathbf{x}' = r\mathbf{R}\mathbf{x} + b\mathbf{b}, \tag{5.2}$$

where r and r' are the depths and b is the baseline scale. When both sides of the equation 5.2 are divided by b, the coupling of the baseline scale b and the depth is evident; in the absence of metric knowledge about r, r' or b, a reconstruction can only be obtained up to an arbitrary scale factor.

Co-planar scene points. Many indoor environments consist of locally planar structures. Let us assume that the camera undergoes a general motion (rotation and translation) and that the scene points observed come from a single plane in the world. The central projections of the co-planar scene points are related by a planar transformation, which can represented by a 3×3 homogeneous *homography* matrix $\mathbf{H}$. Let Π denote a world plane specified in coordinate frame C. The plane is characterized by a unit normal vector $\mathbf{n}$ and distance d. For a scene point X on the plane, the following relationship holds

$$\mathbf{n} \cdot \mathbf{X} = -d. \tag{5.3}$$

In terms of the projection $\mathbf{x}$, equation 5.3 can be written as

$$r(\mathbf{n} \cdot \mathbf{x}) = -d. \tag{5.4}$$

So we have $r = -d/\mathbf{n}^T\mathbf{x}$. If we divide both sides of equation 5.2 by r we obtain

$$\mathbf{x}' \simeq (\mathbf{R} - \frac{b}{d}\mathbf{b}\mathbf{n}^T)\mathbf{x} = \mathbf{H}\mathbf{x}, \tag{5.5}$$

where $\simeq$ denotes similarity up to scale and $\mathbf{H}$ is the homography matrix.

A general homography can be estimated if at least 4 correspondences are known, because each correspondence provides two constraints on the entries of $\mathbf{H}$. Notice that if the camera does not undergo a translation, i.e. b = 0, the homography in equation 5.5 reduces to a rotation matrix. The plane from which a scene point comes is then inconsequential and may be arbitrary.

Arbitrary scene points. If the 3-D scene points observed are in arbitrary positions (i.e. they are scattered in space), and the camera undergoes a general motion, the projections of the scene points are related via the epipolar geometry, which we discussed in chapter 4. The epipolar geometry expresses the fact that for some r and r', the rays, spanned by two corresponding feature points $\mathbf{x} \leftrightarrow \mathbf{x}'$ originating from C and C' respectively, meet at point X in space. The fact that the rays $\mathbf{x}$ and $\mathbf{x}'$ and the baseline $\mathbf{b}$ are co-planar can be expressed as

$$\mathbf{x}' \cdot (\mathbf{b} \times (\mathbf{R}\mathbf{x})) = \mathbf{x}'^T\mathbf{S}\mathbf{R}\mathbf{x} = \mathbf{x}'^T\mathbf{E}\mathbf{x} = 0, \tag{5.6}$$

where × denotes the vector outer product, $\mathbf{S}$ denotes the 3×3 skew symmetric matrix such that $\mathbf{S}\mathbf{x} = \mathbf{b} \times \mathbf{v}$ for any $\mathbf{v}$, and $\mathbf{E} = \mathbf{S}\mathbf{R}$ is the essential matrix. The essential matrix can be estimated using a linear method if at least 8 correspondences are available.

Notice that the essential matrix is degenerate when there is no translation between viewpoints, because then $\mathbf{S}$ becomes a null matrix. A degeneracy also occurs when all correspondence come from a single world plane. In this case, the correspondences are related by an homography, which is uniquely determined by four correspondences. Additional correspondences from the plane do not provide additional constraints on the entries of the essential matrix. In [93] a method is proposed to detect these degeneracies and handle them by switching to homography estimation. In our application we assume that the camera poses associated with any pair of images are always related by a non-zero translation.

Both the homography and the essential matrix encode information about the relative camera poses at which two images are acquired. Several methods exist to recover the motion parameters from an essential matrix [30, 96, 36]. These decompositions yield four possible rotation/translation pairs. The correct decomposition can be selected by reconstructing the 3-D points from the image correspondences according to each pair.

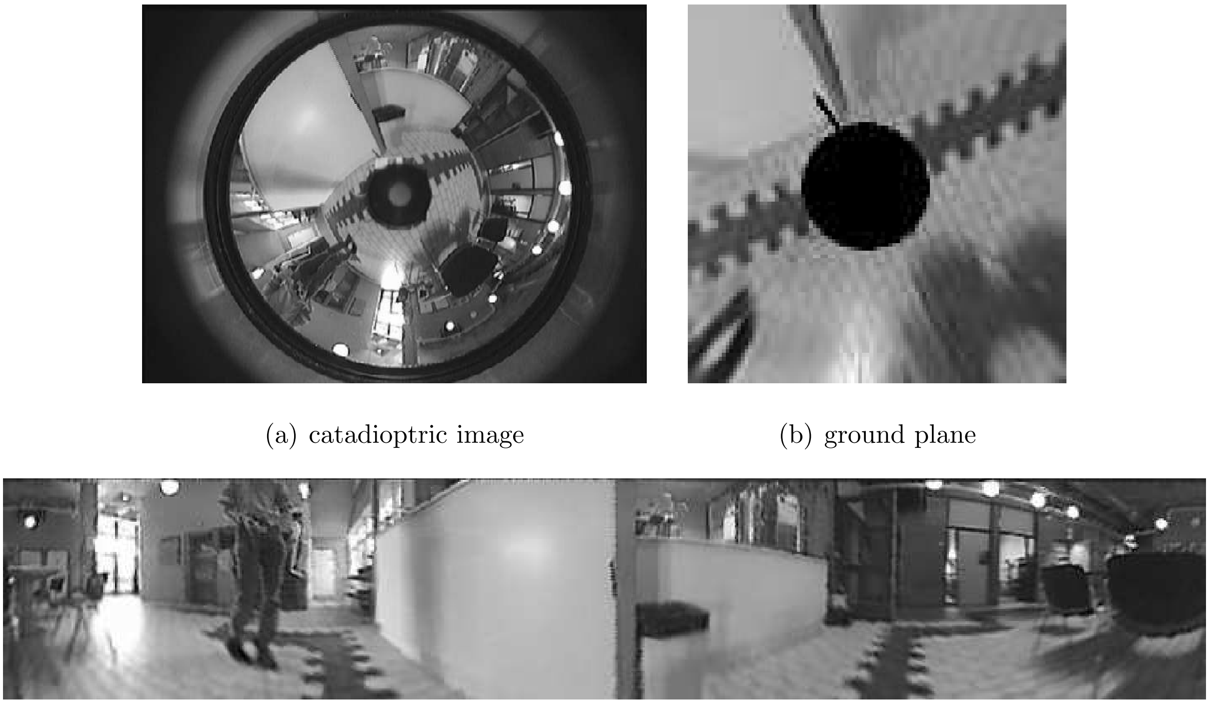
The problem is revisited and explained in more depth in section 5.3.1. The problem of recovering the relative camera poses and plane parameters from a homography is discussed in detail in [18]. In the most general case, eight different sets of solutions exist. However, only two of these are feasible if the world plane is considered to be non-transparent. A method to find these solutions via the SVD decomposition of the homography is presented in [95]. From an estimated homography, the rotation and the product $(b/d)\mathbf{b}\mathbf{n}^T$ are recoverable. If $\mathbf{b}$ and $\mathbf{n}$ are both set to unit length, only the ratio of the actual baseline scale b and the plane distance d can be recovered.

Assuming a general camera motion, we have seen that a reconstruction of the relative camera poses and the structure of the scene can be obtained up to an arbitrary scale factor from both an estimated essential matrix and an estimated homography. From an estimated homography, the ratio of the baseline scale and the plane distance can also be recovered. Suppose now that we have a set of images of a single plane obtained by a camera that moves parallel to the plane, so that the height d of the camera above the plane is fixed. We can then set the plane distance to unit length $d = 1$, so that the baseline scale can be expressed in "camera height units". Provided that the homography relating projections of points on the plane in pairs of images can be estimated, a consistent baseline scale over all input images can then be obtained for each image pair independently. These observations provide the basis on which we build our approach to relative camera pose estimation.

5.3 Two-stage Relative Pose Estimation

Our robot is equipped with a catadioptric vision system, which is mounted vertically on top of the robot. An image acquired by the sensor is shown in figure 5.1a. Because the robot moves in an indoor office environment, it is reasonable to assume that the camera undergoes a planar motion parallel to the ground plane. The height of the camera viewpoint remains fixed and the ground plane is always visible. The fixed height can be used to serve as a yardstick; the baseline scale can be expressed in "camera height" units. The homography relating two images of the ground plane therefore appears to be an ideal candidate to recover relative camera pose from a pair of images. The central issue now becomes how to estimate the homography given a pair of catadioptric images. There are essentially two ways to estimate an homography; one can estimate the homography from feature correspondences, or one can directly register two images by minimizing their intensity discrepancy as a function of the homography parameters.

Evidently, estimation of the homography from image correspondences requires that the correspondences come from the ground plane. In [21] two methods are presented to detect a dominant plane on the basis of feature points extracted in two images. The first method assumes that a set of image correspondences, possibly contaminated by outliers (erroneously matched points), is available a priori. A robust estimator is then used to find the homography consistent with the largest subset of correspondences. The second method does not assume that correspondences are known a priori. Instead, correspondences and



(a) catadioptric image

(b) ground plane

(c) cylindrical panoramic image

Figure 5.1: Various image representations. a) catadioptric image, b) ground plane image obtained by re-projecting the catadioptric image onto a plane parallel to the floor, c) cylindrical panoramic image obtained by re-projecting the catadioptric image onto a cylinder.

the homography are solved simultaneously by a robust estimator. The estimator selects four feature points in the first image at random. For each selected feature point in the first image, a feature point in the second image is selected as a potential correspondence. The selection criterion is based on image similarity between windows centered at the feature point in the first image and windows centered at the feature points in the second image. The four points in the first image and the four points in the second image uniquely determine an homography. The "generated" homography is evaluated by warping the feature points in the first image to the second image and calculating an evaluation measure as follows. For each warped feature point, the nearest feature point in the second image is considered to be the corresponding feature. The evaluation measure combines the distances to the corresponding features and the image similarity between corresponding features. The procedure is performed many times while storing the homographies and their respective evaluation. Finally, the homography yielding the best evaluation is considered to be the homography describing the dominant plane in the scene.

In [52] a two stage procedure is presented to detect a dominant plane on the basis of feature correspondences across two un-calibrated perspective images. The first stage estimates the epipolar geometry relating the two images using a robust estimator. The estimator used is the Least of Median Squares (LMedS)estimator [78]. LMedS aims at estimating the a set of model parameters that fit the majority of the measurements using a Monte-Carlo technique. The idea is to repeatedly draw random subsets from the set of measurements, each subset being just large enough to *calculate* the model parameters. The model consistent with the majority of the measurements is retained as the solution. The estimated epipolar geometry, which is assumed to be accurate, provides six linear constraints on any homography. Thus, two pairs of corresponding points from a single plane suffice to derive the homography. In order to find the dominant plane, again an LMedS based procedure is used.

The approaches described above assume that a large set of correspondences (which may be contaminated by outliers) is available. From images of the ground plane, a large set of point correspondences may be hard to obtain for various reasons. First, the ground plane itself may contain few easily identifiable feature points (such as corners). Second, a rotation invariant descriptor of feature points should be used because a rotation of the robot causes a rotation of the image. Such invariance is important to establish correspondences, but always comes at the price of less discriminative power. It is therefore questionable that the ground plane will be identifiable as the dominant plane using the above methods. We propose a different approach in which we take advantage of our catadioptric vision sensor. Similar to [52] our approach is a two-stage estimation procedure in which we first estimate the epipolar geometry relating a pair of images and then use the resulting estimate to constrain the homography relating ground plane images.

In the first stage, we estimate the essential matrix from point correspondences established between *cylindrical panoramic images* such as shown in figure 5.1c. The cylindrical panoramic image is obtained by re-projecting the catadioptric image shown in figure 5.1a onto a virtual cylinder. Corresponding feature points established in cylindrical panoramic images are likely to come from scene points in arbitrary positions, a prerequisite to estimate the essential matrix. Due to the panoramic field of view of the images, many easily identifiable features can be found. Furthermore, features remain visible after rotation and after relatively large camera displacements (unless they get occluded by other objects). When cylindrical panoramic images are used, off-the-shelf tracking techniques (e.g. [3]), originally developed for conventional camera's, can be employed to reliably establish feature correspondences. The relatively few tracking errors that occur are easily dealt with by robust statistical techniques. It has been shown in several works [86, 50, 28] that, in spite of the relatively low resolution, catadioptric panoramic vision can produce more accurate camera motion estimates than those obtained from conventional cameras. In particular, small rotations of the sensor can better be distinguished from small translations. From the estimated essential matrix we derive an estimate of the relative rotation $\mathbf{R}$ and baseline direction $\mathbf{b}$.

In the second stage, we use the estimated rotation and baseline direction to constrain the homography relating the ground plane in two *ground plane images*. A ground plane image, as shown in figure 5.1b, is obtained from the catadioptric image, as shown in figure 5.1a, by re-projecting the catadioptric image onto a virtual plane parallel to the ground plane. The single parameter left to relate the ground plane images is related to the baseline scale b. As explained before, extraction of reliable feature correspondences coming from the ground plane may be difficult. Therefore, instead of pursuing a feature correspondence based approach, we attempt to register the ground plane images as a function of the baseline scale b by minimizing the intensity discrepancy between two images.

5.3.1 Estimating the rotation and baseline direction

In this section we describe a method to recover the relative rotation and direction of translation from a pair of cylindrical panoramic images. We adopt a commonly used approach in which the relative camera poses are derived via decomposition of an estimated essential matrix. The essential matrix is estimated using a robust linear estimator.

We exploit prior knowledge that the motion of the camera is parallel to the ground plane. If the camera is mounted vertically on top of the robot, the camera is restricted to a translation parallel to the camera X-Y-plane, and a rotation of the camera is about an axis parallel to the camera Z-axis. A novelty in our method is that we explicitly incorporate this prior knowledge in the parameterization of the essential matrix.

Parameterization of the essential matrix. We set the baseline to unit length, such that it can be specified by $\mathbf{b} = [\cos\phi, \sin\phi, 0]^T$. The rotation is a rotation through an angle θ about the Z-axis. The essential matrix then has the following form:

$$\begin{aligned} \mathbf{E} &= \mathbf{SR} \\ &= \begin{bmatrix} 0 & 0 & \sin\phi \\ 0 & 0 & -\cos\phi \\ -\sin\phi & \cos\phi & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & \sin\phi \\ 0 & 0 & -\cos\phi \\ \sin(\theta-\phi) & \cos(\theta-\phi) & 0 \end{bmatrix}. \end{aligned} \tag{5.7}$$

We refer to essential matrices of this form as *specialized* essential matrices.

Estimation of the specialized essential matrix. We adopt a robust linear estimation method to estimate the essential matrix. If we rewrite the specialized essential matrix as a general essential matrix, we obtain

$$\mathbf{E} = \begin{bmatrix} 0 & 0 & e_3 \\ 0 & 0 & e_6 \\ e_7 & e_8 & 0 \end{bmatrix}. \tag{5.8}$$

The essential matrix can be estimated from a set of N correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ ($i \in \{1, \ldots, N\}$) as follows. If the non-zero entries of the specialized essential matrix are expressed as a vector $\mathbf{e} = [e_3, e_6, e_7, e_8]^T$, the N constraints obtained from N correspondences can be expressed linearly as $\mathbf{De} = 0$, where $\mathbf{D}$ is the $(N \times 4)$ design matrix. Each row $\mathbf{D}_i$ of $\mathbf{D}$ has the form:

$$\mathbf{D}_i = \begin{bmatrix} \mathbf{x}'_i(1)\mathbf{x}_i(3) & \mathbf{x}'_i(2)\mathbf{x}_i(3) & \mathbf{x}'_i(3)\mathbf{x}_i(1) & \mathbf{x}'_i(3)\mathbf{x}_i(2) \end{bmatrix}. \tag{5.9}$$

A solution for $\mathbf{e}$ can be found linearly by solving

$$\min_{\mathbf{e}} \|\mathbf{De}\|^2 \quad \text{subject to} \quad \|\mathbf{e}\| = 1. \tag{5.10}$$

The constraint $\|\mathbf{e}\| = 1$ is incorporated to fix the scale of $\mathbf{E}$. The solution to this problem is the eigenvector of the moment matrix $\mathbf{M} = \mathbf{D}^T\mathbf{D}$ associated with the smallest eigenvalue and can be found by using a singular value decomposition (SVD) of $\mathbf{D}$. This algorithm is known as the 8-point algorithm [31].

An essential matrix has two equal eigenvalues and has rank two. The linear 8-point algorithm does not enforce these properties on the recovered $\mathbf{E}$ matrix. The nearest essential matrix (Frobenius norm) can be found as follows. Let $\hat{\mathbf{E}}$ denote the matrix found by the 8-point algorithm. Let its SVD decomposition be $\hat{\mathbf{E}} = \mathbf{USV}^T$ where $\mathbf{S} = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$. The nearest essential matrix $\mathbf{E}$ can the be found as $\mathbf{E} = \mathbf{US}'\mathbf{V}^T$, where $\mathbf{S}' = \text{diag}((\sigma_1 + \sigma_2)/2, (\sigma_1 + \sigma_2)/2, 0)$.

Image correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ are typically established by image based matching and are inherently noisy. Furthermore, erroneous associations between image features may be present in the set of correspondences. Erroneous associations act as outliers that do not adhere to the relationship implied by the essential matrix. The linear 8-point algorithm is sensitive to noise and cannot cope with outliers [94]. Robust statistical estimation techniques can be employed to identify and discard such outliers from the set of correspondences. We first use a Least of Median Squares (LMedS) estimator [78] to obtain an initial estimate of the essential matrix.

Because the solution is based on a minimal number of measurements, it can be improved by re-estimating the model from the set of measurements consistent with the proposed solution. This set may still contain a number of outliers. In order to identify and discard the remaining outliers, we apply an M-estimator. An M-estimator assigns low weights to outlying measurements so that their influence is reduced [114]. We implemented the M-estimator as an iteratively re-weighted least squares variant of the linear least squares method implied by the 8-point algorithm [94].

Recovering the relative camera poses. Estimation of the specialized matrix is a means to an end. We are not interested in the entries of the estimated essential matrix per se, but in the relative poses characterized by the baseline direction $\mathbf{b}$ (parameterized by ϕ) and the rotation $\mathbf{R}$ (parameterized by θ). Many decomposition techniques exist to recover the rotation and baseline direction from an essential matrix. We propose a new method tailored to the specialized essential matrix to recover ϕ and θ directly.

The baseline direction ϕ can be obtained as

$$\phi = \arctan2\left(\frac{\sin\phi}{\cos\phi}\right) = \arctan2\left(\frac{e_3}{-e_6}\right), \tag{5.11}$$

where $\arctan2(y/x)$ is the four quadrant arctangent of x and y so that $-\pi \leq \arctan2(y/x) \leq \pi$.

Because of sign ambiguity, there are two possible solutions for the baseline:

$$\mathbf{b} = [\cos\phi, \sin\phi, 0]^T, \tag{5.12}$$

$$\mathbf{b}_\pi = [\cos(\phi+\pi), \sin(\phi+\pi), 0]^T. \tag{5.13}$$

The camera rotation angle θ can be recovered from the last row of $\mathbf{E}$ as

$$\theta == (\theta - \phi) + \phi = \arctan2\left(\frac{e_7}{e_8}\right) + \arctan2\left(\frac{e_3}{-e_6}\right). \tag{5.14}$$

The rotation matrix, $\mathbf{R}$, associated with baseline direction $\mathbf{b}$ is then given by

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.15}$$

The rotation matrix, $\mathbf{R}_\pi$, associated with baseline direction $\mathbf{b}_\pi$ is related to $\mathbf{R}$ by a rotation π about an axis parallel to $\mathbf{b}$. We derive $\mathbf{R}_\pi$ using Rodrigues formula for rotations [23]:

$$\mathbf{R}(\mathbf{v},\alpha) = \cos\alpha \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \sin\alpha \begin{bmatrix} 0 & -\mathbf{v}(3) & \mathbf{v}(2) \\ \mathbf{v}(3) & 0 & -\mathbf{v}(1) \\ -\mathbf{v}(2) & \mathbf{v}(1) & 0 \end{bmatrix} + (1-\cos\alpha)\mathbf{v}\mathbf{v}^T, \tag{5.16}$$

which gives the rotation matrix corresponding to a rotation α about a unit norm vector $\mathbf{n}$. For a rotation π about $\mathbf{b} = [\cos\phi, \sin\phi, 0]^T$ equation 5.16 gives

$$\mathbf{R}(\mathbf{b},\pi) = \begin{bmatrix} 2\cos^2\phi - 1 & 2\cos\phi\sin\phi & 0 \\ 2\cos\phi\sin\phi & 2\sin^2\phi - 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \tag{5.17}$$

The rotation matrix $\mathbf{R}_\pi$ is then given by:

$$\mathbf{R}_\pi = \mathbf{R}\mathbf{R}(\mathbf{b},\pi). \tag{5.18}$$

The co-planarity constraint implemented by a given essential matrix holds for any non-zero scalar multiple of that matrix. In particular, if $\mathbf{E}$ is a solution, so is $-\mathbf{E}$. Matrix $-\mathbf{E}$ has two decompositions that are related to those already found: $\mathbf{b}_\pi$ with orientation $\mathbf{R}$ and $\mathbf{b}$ with orientation $\mathbf{R}_\pi$. Thus, there are four possible pairs of rotation and translation, each resulting in an essential matrix consistent with the set of feature correspondences.

Selection of rotation and translation. To resolve the ambiguity one can go back to the 3-D space and determine the point where two corresponding rays meet. Figure 5.2 displays top views of cylindrical camera pairs and illustrates the geometric interpretation of the four possible combinations of rotation and translation. A $+$ symbol indicates a positive depth and a $-$ symbol indicates a negative depth. The correct rotation/translation pair ideally holds positive depths from both viewpoints for all corresponding features used to estimate the essential matrix.

If we divide both sides of equation 5.2 by r and rearrange the terms then for any correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ we have

$$(r'/r)\mathbf{x}' - (1/r)\mathbf{b} = \mathbf{R}\mathbf{x}. \tag{5.19}$$

In order to recover r and r', a least squares problem of the form $\mathbf{A}\mathbf{p} = \mathbf{c}$ can be formulated as

$$\begin{bmatrix}\mathbf{x}' & -\mathbf{b}\end{bmatrix}\begin{bmatrix}r'/r \\ 1/r\end{bmatrix} = \begin{bmatrix}\mathbf{R}\mathbf{x}\end{bmatrix}. \tag{5.20}$$

A solution can be obtained using the pseudo inverse technique [73]:

$$\mathbf{p} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{c}. \tag{5.21}$$

We call the solution positive when both $\mathbf{p}(1) > 0$ and $\mathbf{p}(2) > 0$. Otherwise, a solution is called negative. In practice, correspondences are noisy. As a result, even if the correct rotation/translation pair is selected, not every solution is positive. In particular, rays nearly parallel to the baseline may yield a negative solution due to small errors in the estimated baseline direction. Furthermore, distant 3-D points give rise to nearly parallel rays from both viewpoints. The intersection of the back-projected image coordinates $\mathbf{x}$ and $\mathbf{x}'$ is ill determined (can be +far or -far). In order to cope with these issues, we adopt a majority voting scheme. For each of the four possible rotation/translation pairs we count the number of positive solutions and select the pair yielding the highest number of positive solutions.

5.3.2 Estimating the baseline scale

Assuming that $\mathbf{R}$ and $\mathbf{b}$ have been recovered from the essential matrix, we estimate the baseline scale b as follows. The catadioptric image is re-projected onto a plane parallel to the floor. The planar perspective projections of points on the ground plane are related by a homography. We assume that the ground plane normal $\mathbf{n}$ is parallel to the camera Z-axis, which coincides with the axis of camera rotation, and that the camera translation is in the camera $X-Y$ plane. The homography (whose general form is given in equation 5.5), then takes the form of a 2-D rigid transformation:

$$\mathbf{H} = \begin{bmatrix}\cos\theta & -\sin\theta & -(b/d)\cos\phi \\ \sin\theta & \cos\theta & -(b/d)\sin\phi \\ 0 & 0 & 1\end{bmatrix}, \tag{5.22}$$

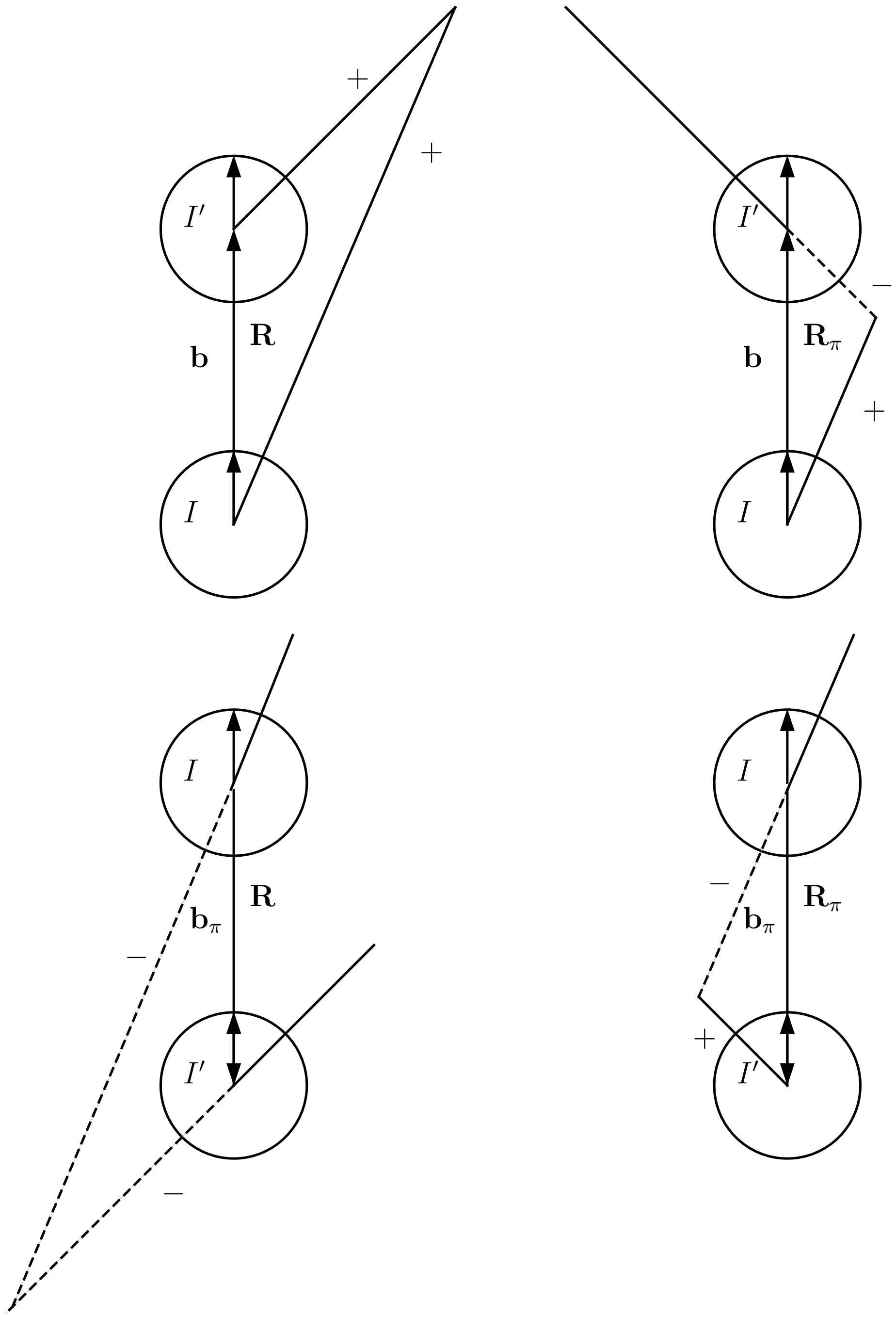


Figure 5.2: Top view of the four possible combinations of rotation and translation that can be derived from an essential matrix. The ambiguity can be resolved by considering visibility of the feature correspondences from which the essential matrix was estimated. The correct rotation/translation pair yields a positive depth (marked by +'s) from viewpoints I and I' for most of the features that were used to estimate the essential matrix.

where θ and ϕ describe the rotation and baseline direction recovered from the previously estimated essential matrix. We set $d = 1$, leaving the baseline scale b (which is expressed in "camera height" units) as the only free parameter to register projections of points on the ground plane.

In order to estimate b, we register the ground plane images by minimizing the squared intensity difference between the two images as a function of b. The camera calibration matrix $\mathbf{K}$ of the virtual perspective camera is known. Pixel coordinates $\mathbf{u} = \mathbf{Kx}$ and $\mathbf{u}' = \mathbf{Kx}'$ in two images are then related via the homography

$$\mathbf{u}' = \mathbf{KHK}^{-1}\mathbf{u}. \tag{5.23}$$

We then minimize the following error function

$$E^2 = \sum_i \left[I'(\mathbf{u}_i')) - I(\mathbf{u}_i)\right]^2 = \sum_i e_i^2, \tag{5.24}$$

where i is a pixel index, I and I' are images and $\mathbf{u}_i' = \mathbf{KH}(\mathbf{R}, \mathbf{n}, d, \mathbf{b}, b)\mathbf{K}^{-1}\mathbf{u}_i$. The error function is minimized as a function of b (the other parameters are kept fixed) by iteratively updating the estimate of b as follows. Given a current estimate of b, the error function is linearized as

$$E \approx \sum_i \left[e_i(b) + \frac{\partial e_i}{\partial b}\Delta b\right]^2. \tag{5.25}$$

We explain how we obtain $\partial e_i/\partial b$ later. For now, assume that the partial derivatives are available. Gradient descend of the error function with respect to Δb can be performed by setting

$$\begin{aligned} 0 = \frac{\partial E}{\partial \Delta b} &\approx \frac{\partial}{\partial \Delta b}\sum_i \left[e_i(b) + \frac{\partial e_i}{\partial b}\Delta b\right]^2 \\ &= 2\sum_i \left[\left(\frac{\partial e_i}{\partial b}\right)^2 \Delta b + \left(\frac{\partial e_i}{\partial b}\right) e_i\right]. \end{aligned} \tag{5.26}$$

This can be rewritten as an equation of the form $a\Delta b = c$, where

$$a = \sum_i \left(\frac{\partial e_i}{\partial b}\right)^2 \quad \text{and} \quad c = -\sum_i e_i \left(\frac{\partial e_i}{\partial b}\right). \tag{5.27}$$

The solution for Δb is then given by $\Delta b = c/a$. The model parameter b can now be updated as $b \leftarrow b + \Delta b$. These steps can be repeated until the error stabilizes or until a maximum number of iterations is reached. Instead of straightforward gradient, we use the more efficient Levenberg-Marquardt (LMQ) method [73]. [1]

[1] The Levenberg-Marquard method solves the equation $(a + \sigma)\,\Delta b = c$, where σ is a time varying stabilizing parameter. The value of σ is initialized to a small value. At the end of each iteration, if the update Δb found reduces the error, the update is accepted and σ is decremented. On the other hand, if the update leads to error increase, σ is incremented and the equation is solved for again. This step is then repeated until an update is obtained that reduces the error, which is bound to happen because for large σ the method approaches a steepest descent.

At each iteration an estimate of the gradient of the error function with respect to b is required. By application of the chain rule, the gradient can be calculated as

$$\frac{\partial e_i}{\partial b} = \frac{\partial I'(\mathbf{u'}_i)}{\partial \mathbf{u'}_i} \frac{\partial \mathbf{u'}_i}{\partial \mathbf{h}} \mathbf{K} \frac{\partial \mathbf{H}}{\partial b} \mathbf{K}^{-1}, \tag{5.28}$$

where $\mathbf{h}$ is a 9×1 vector containing the elements of the current estimate of $\mathbf{KHK}^{-1}$. We obtain the partial derivatives of I' with respect to the image coordinates $\mathbf{u}'$ a regularized image gradient operator [77]. The image coordinates $\mathbf{u}'$ generally do not correspond to discrete pixel coordinates and a form of interpolation is required to perform a re-sampling of image I'. We use bi-linear interpolation. The partial derivatives of $\mathbf{u}'$ with respect to $\mathbf{h}$ are given by

$$\frac{\partial \mathbf{u}'}{\partial \mathbf{h}} = \begin{bmatrix} \mathbf{u}(1)D & \mathbf{u}(2)D & 1 & 0 & 0 & 0 & -\mathbf{u}(1)\mathbf{u}'(1)D & -\mathbf{u}(2)\mathbf{u}'(1)D & -\mathbf{u}'(1)D \\ 0 & 0 & 0 & \mathbf{u}(1)D & \mathbf{u}(2)D & 1 & -\mathbf{u}(1)\mathbf{u}'(2)D & -\mathbf{u}(2)\mathbf{u}'(2)D & -\mathbf{u}'(2)D \end{bmatrix}, \tag{5.29}$$

where $D = 1/(\mathbf{h}(7)\mathbf{u}(1) + \mathbf{h}(8)\mathbf{u}(2) + \mathbf{h}(9))$. Finally, referring to the definition of $\mathbf{H}$ in equation 5.22, $\partial\mathbf{H}/\partial b$ is given by

$$\frac{\partial \mathbf{H}}{\partial b} = \begin{bmatrix} 0 & 0 & -\cos\theta \\ 0 & 0 & -\sin\theta \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.30}$$

As any non-linear optimization technique, the LMQ method may get stuck in a local minimum of the error function. To diminish the risk of getting stuck, the error function can be smoothed by computing image gradients at a large scale. A drawback of this approach is that the minimum can be located less accurately. Instead, we precede the LMQ method by generating a small set of proposal baseline lengths. Each proposed b yields an homography that is used to calculate the corresponding value of the error function E. Subsequently, the proposed b yielding the smallest error is used as a starting point for the LMQ minimization.

The intensity error minimization procedure minimizes the total intensity error over *all* overlapping pixels in the two images that are registered. Ideally however, only “floor pixels” (projection of points on the ground plane) should participate in the minimization. The reason is that the method assumes that the motion of all points in the image can be described by a single homography. For non-floor points (like extruding objects and specular reflections) an additional image motion component is induced by a camera motion that depends on their height above the plane. Automatically deciding which pixels are “floor” and which are “non-floor” is a non-trivial issue. If the homography were known, pixels may be classified as “floor” or “non-floor” pixels on the basis of the intensity difference between the target image $I(\mathbf{u})$ and the warped source image $I(\mathbf{u}')$. This approach is commonly used to detect obstacles on a ground plane [25, 52]. However, we initially have a rough estimate of the homography. Because the images are not registered accurately, it makes little sense to employing such a classification scheme. Typically, intensity based error minimization ignores the issue and implicitly assumes that all pixels come from the same, non-specular, world plane. We are willing to adopt the same approach, but only after verifying experimentally that the baseline length estimate is not affected much by extruding objects and specular reflections.

5.4 Experiments

Ground plane images that are used to estimate the baseline length may display objects extruding from the floor. Furthermore, the floor may give rise to specular reflections. Allowing objects other than the floor and specular reflections to contribute in the estimation of the ground plane homography may affect the estimated baseline length. In section 5.4.1 we investigate experimentally how the baseline length estimate is affected by extruding objects and illumination effects. In section 5.4.2 we use our technique to estimate relative camera poses on images acquired by a real robot. We choose to apply our technique to subsequent image pairs, taken from a sequence of images acquired during robot navigation. The relative pose estimates obtained are composed to reconstruct the trajectory traversed by the robot. Applied in this fashion, our approach yields a form of visual odometry.

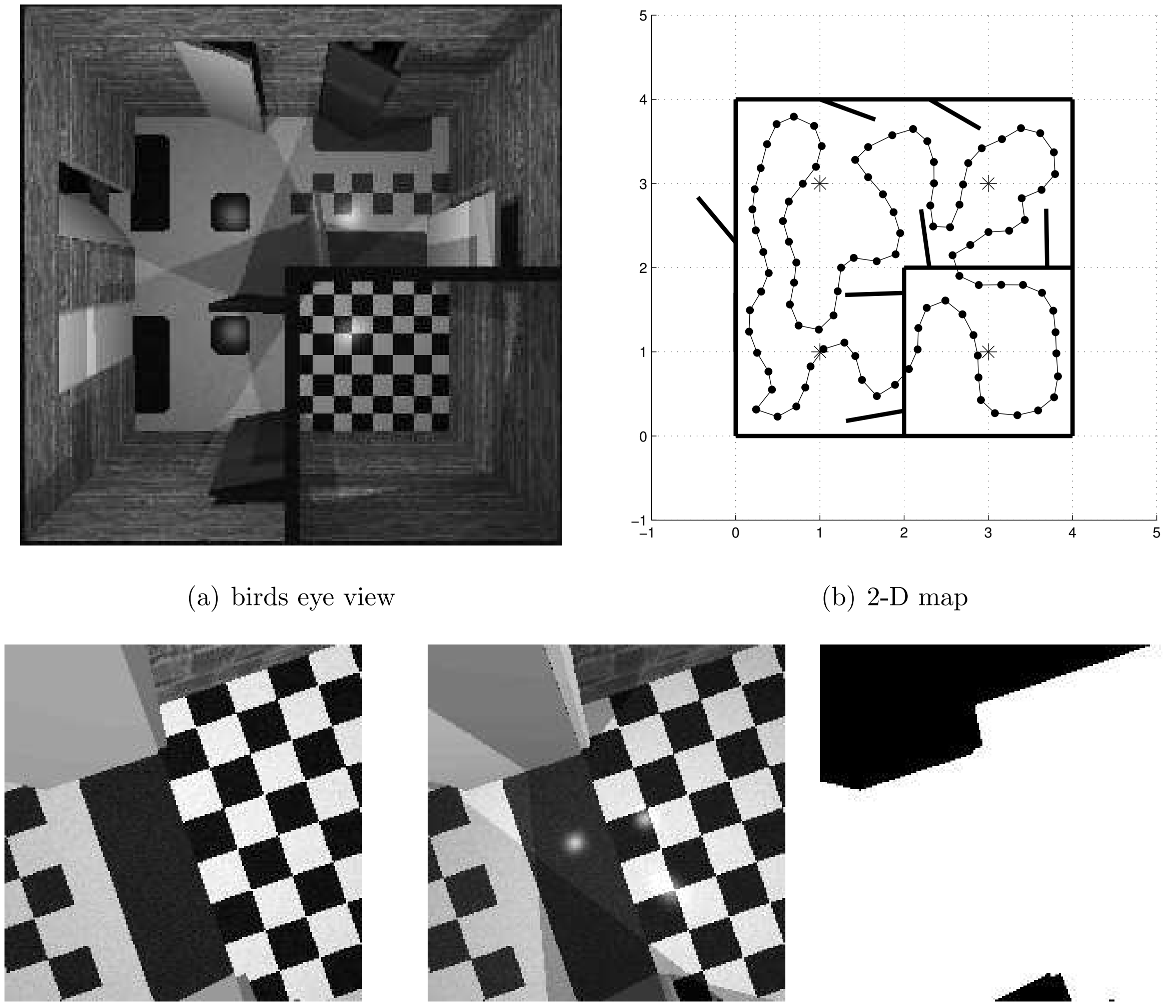
5.4.1 Simulation experiments

In a series of simulation experiments we investigated the sensitivity of stage two of our approach (estimation of the baseline scale) with respect to objects extruding from the ground plane and specular reflections on the ground plane.

Data and method. We designed a virtual environment covering an area of 4×4 units (think of units as meters). A rendered birds-eye view and the 2-D map of the environment are show in figures 5.3a and b. The 2-D map shows the location of walls, doors, light sources and a simulated camera trajectory through the environment. Images are rendered by a ray-tracer that simulates a camera aimed straight at the ground plane at 98 camera poses along the trajectory. The height of the camera remains fixed at 0.6 units above the ground plane. Each image is 200×200 pixels and covers a field of view of $120°$ both in horizontal and vertical direction.

Two sets of images were rendered:

"nolights". Images in this set display only the intrinsic color of observed objects in the environment. The images are unaffected by illumination conditions. An example image is shown in figure 5.3c.



(a) birds eye view

(b) 2-D map

(c) “nolights”

(d) “lights”

(e) mask

Figure 5.3: Simulation data. a) A rendered birds-eye overview of the virtual environment. b) A 2-D map of the virtual environment. The asterisks indicate the positions of lights, which are mounted in the ceiling. The curve represents the trajectory travelled by a virtual robot, and the dots on the curve mark the positions where images are obtained. c) An image from the set “nolights”, which is unaffected by illumination effects. d) The corresponding image from the set “lights”. Notice the presence of specular reflections and shadows. e) The corresponding binary mask image indicating which pixels come from extruding objects (black) and from the floor (white).

"lights". Images in this set were rendered using a more realistic illumination model (adopted from [20]). They display both specular reflections and shadows. An example image is shown in figure 5.3d.

For each image, we also rendered a binary mask image, such as shown in figure 5.3e. Pixels in a mask image are white if they come from the floor, and black if they come from to an extruding object such as a wall or door.

The baseline length between pairs of consecutive images in the sequence is 0.25 units. The i-th pair consists of image I_i (source image) and I_{i+1} (target image) ($1 \leq i < 97$). Because we used a simulation environment, the homography relating any pair of images is known exactly. In order to quantify the fraction of non-floor pixels in a pair of images, we warped the source image to the target image according to the exact homography. The warped source image partially overlaps the target image. The fraction of non-floor pixels is now defined as the number of non-floor pixels in the warped source image divided by the total number of pixels in the overlapping part. The largest fraction of non-floor pixels is 57.3%. The smallest fraction of non-floor pixels is 3.4%. On average 26.7% of the all pixels in the sequence are non-floor pixels with a standard deviation of 14.7%.

In order to investigate the influence of extruding objects and specular reflections we devised the following experiment. Given a pair of images, let the true homography relating the images be denoted by $\mathbf{H}^*(\mathbf{K}, \mathbf{R}, \mathbf{n}, d, \mathbf{b}, b^*)$. We generate an homography corresponding to an error Δb in the baseline scale $\mathbf{H}(\mathbf{K}, \mathbf{R}, \mathbf{n}, d, \mathbf{b}, b)$, where $b = b^* + \Delta b$. In our experiments, Δb was drawn randomly from a uniform distribution in the range $[-0.2, 0.2]$. The resulting homography is used as a starting point for the intensity based error minimization. After convergence, the minimization yields an estimated homography $\mathbf{H}(\mathbf{K}, \mathbf{R}, \mathbf{n}, d, \mathbf{b}, \hat{b})$. The error $e = b^* - \hat{b}$ between the true baseline length and the estimated baseline length, is used as a measure to characterize the accuracy of the estimated baseline length. We repeated this experiment 25 times for each of the 97 image pairs, using a different initial homography in each experiment.

Using the binary mask images to prohibit pixels coming from extruding objects from contributing in the intensity error minimization, the following experiments were done and the obtained results are compared.

Experiment 1 ("nolights", "mask"). Images from the set "nolights" were used. Pixels coming from objects extruding from the floor were prohibited from participating in the intensity error minimization.

Experiment 2 ("nolights", "nomask"). Images from the set "nolights" were used. All pixels participated in the intensity error minimization.

Experiment 3 ("lights", "mask"). Images from the set "lights" were used. Pixels coming from objects extruding from the floor were prohibited from participating in the intensity error minimization.

	nomask	**mask**
nolights	0.0025	0.0027
lights	0.0035	0.0045

Table 5.1: Median of absolute baseline length errors over all image pairs for each data set after optimization. The average baseline length between subsequent poses in the data set 0.25 units.

Experiment 4 ("lights", "nomask"). Images from the set "lights" were used. All pixels participated in the intensity error minimization.

The influence of extruding objects was investigated by comparing the accuracy of the baseline estimation obtained in experiment 1 and experiment 2. Observed differences in results from the two experiments (if any) can primarily be contributed to the non-floor pixels.

Images in the "lights" exhibit specular reflections on the ground plane. Such reflection appear to be moving along with the observer, and therefore do not adhere to the homography relating points on the ground plane. The influence of specular reflections was investigated by comparing the accuracy of the baseline estimation obtained in experiment 1 and experiment 3, and by comparing the results from experiment 2 and experiment 4. Observed differences in results (if any) can primarily be contributed primarily to the presence of specular reflections.

Results. The results of the four experiments are summarized in table 5.1, which presents the median of the absolute baseline length errors for all four conditions. The median was used instead of the mean because the non-linear method used to estimate the baseline length did not always converge to a minimum close to the true baseline length (which is 0.25 units in our experimental setup).

We first observe that the influence of shadows and specular reflections is more notable than the influence of extruding objects. The expected baseline length error increases when realistic illumination conditions are simulated.

This could be expected because specular reflections are typically seen as bright spots on the plane. These spots yield a strong image gradient, which has a large influence on the intensity minimization.

It is interesting to note that it appears to be better not mask out non-floor pixels, particularly under realistic illumination conditions ("lights" data set). An plausible explanation is that the gradient information at the boundaries between floor and extruding objects contributes to finding a good solution. Gradient information from objects extruding from the floor is not available when the binary mask images are used. The influence of specular reflections may even be compensated for by this gradient information.

In a way, this is good news because it implies that we do not need to worry about extruding objects when estimating the baseline scale. It may be argued that images

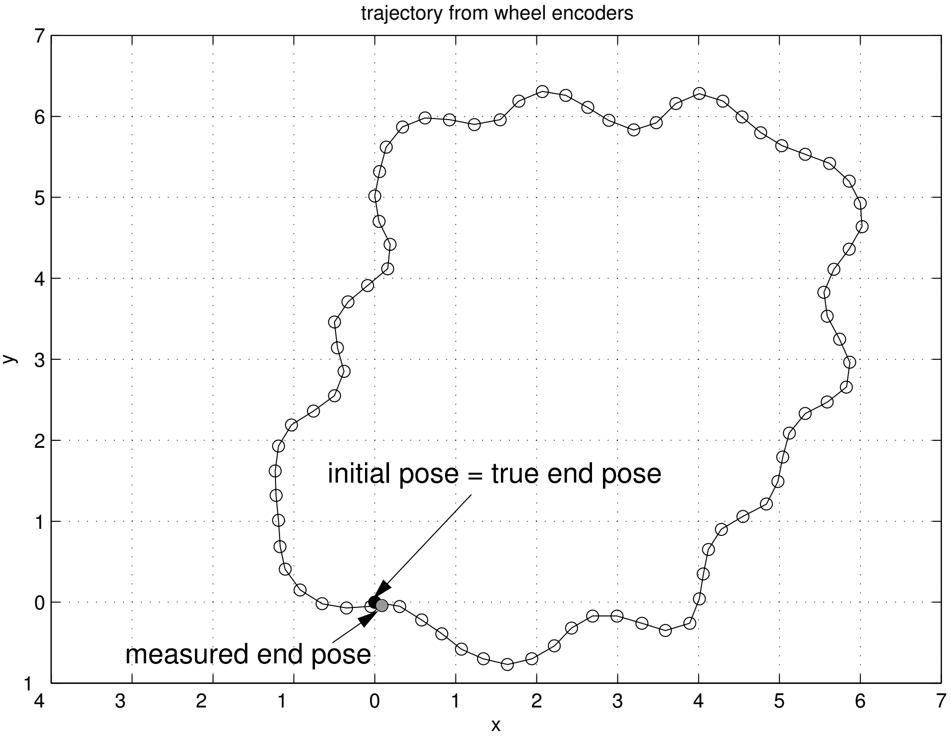


Figure 5.4: Trajectory as measured by the robot's wheel encoders. The closed-loop trajectory starts at (0,0). At the end of the trajectory, the robot arrived exactly at the start pose, but wheel odometry shows a slight mismatch between the start pose and end pose.

should have been used that contain an even larger fraction of non-floor pixels (than the 57.3% in our data sets). We believe however that the camera trajectory is fairly representative for a trajectory real mobile robot could traverse.

5.4.2 Visual odometry experiments

If a sequence of images ${}^{0}I, \ldots, {}^{N}I$ is available, each image ${}^{j}I$ $(1 < j \leq N)$ can be related to the first image in the sequence (which serves as a reference frame) by multiplying homographies relating consecutive images:

$$ {}^{0}\mathbf{x} = \left(\prod_{i=1}^{j} {}^{i-1}_{i}\mathbf{H} \right) {}^{j}\mathbf{x}, \qquad (5.31)$$

resulting in a form of visual odometry. In this section we apply such visual odometry to reconstruct the trajectory traversed by our robot while capturing images.

Experimental setup. A sequence of catadioptric images was acquired while manually guiding our robot through our building. The trajectory described a closed loop where the robot's initial pose and the pose at the end of the trajectory are identical. The trajectory

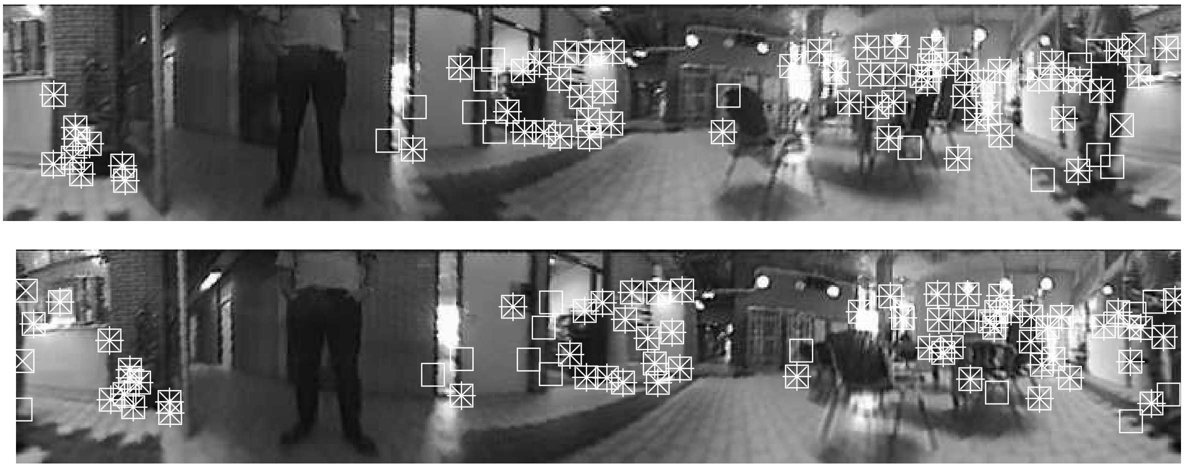


Figure 5.5: Correspondences obtained by feature tracking and subsequent robust estimation of the essential matrix. Boxes represent correspondences obtained by the feature tracker. Boxes with a × symbol represent correspondences that are compatible with an estimate of the essential matrix obtained by a Least of Median Squares estimator. Boxes with a × and a + denote correspondences compatible with an estimate of the essential matrix obtained by an M-estimator.

as measured by the robot's wheel encoders is displayed in figure 5.4. Over the relatively short trajectory, without any sudden sharp turns or bumps, odometry was quite accurate. A total of 81 600×450 pixel catadioptric images were captured during navigation. The derived cylindrical panoramic images are 720×120 pixels. The derived ground plane images are 200×200 pixels. The virtual perspective camera observing the ground plane was given a field of view of 140°.

Estimation of the essential matrix. The Kanade-Lucas-Tomasi feature tracker [45, 3] (KLT) was used to track a set of at most 200 salient image features from frame to frame. Lost features are replaced immediately by new salient features. Features tracked successfully between two frames are used to estimate the essential matrix relating the two frames. First, the Least of Median Squares (LMedS) method [78] is used to identify and discard gross outliers. Subsequently, an M-estimator [94, 114] is used to refine the estimate.

Figure 5.5 shows an example of the results of tracking and essential matrix estimation. Boxes indicate points that were marked as successfully tracked by KLT. Boxes with a × indicate correspondences classified as inliers after LMedS. Boxes with both a × and a + indicate surviving points after applying LMedS followed by the M-estimator. Notice that most erroneously tracked points are correctly identified by the estimation procedure. Also notice that only a very small fraction of salient points lie on the ground plane.

Estimation of the baseline scale. The robot is always visible in the ground plane images. Prior to intensity minimization, it was masked out so that the image of the robot does not contribute in the estimation of the scale. In this experiment, the scale at which

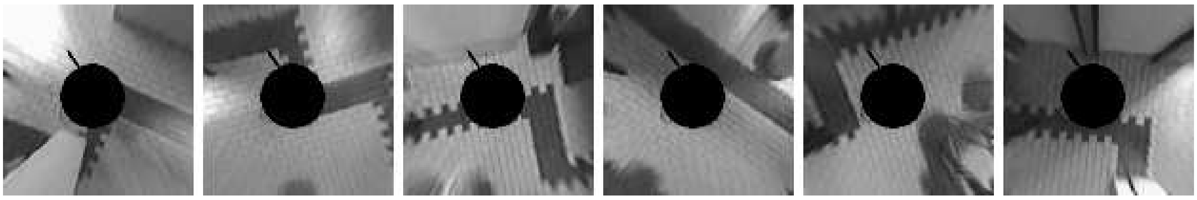


Figure 5.6 : A random selection of ground plane images from the sequence.

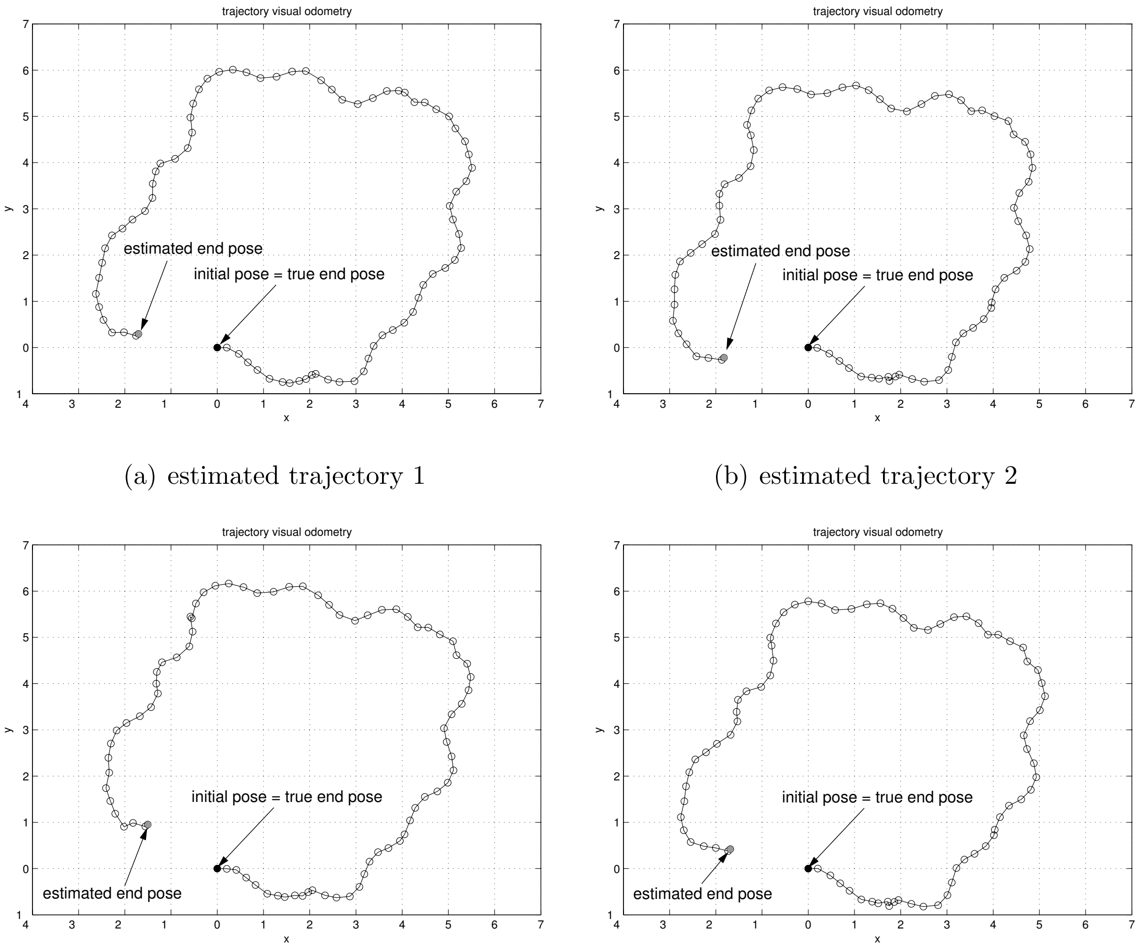
image gradients were computed were set to 1 pixel. The initial sampling of b values was done in the range $[0, 1.2]$. The limits were chosen because negative b values corresponds to a baseline in the opposite direction and for $b > 1.2$ there was no overlap between a warped source and target image. A random selection of ground plane images used in the experiment is shown in figure 5.6.

Visual odometry results. Figure 5.4 displays the trajectory as measured by the robot wheel encoders. Some reconstructed trajectories obtained using our vision based method are shown in figure 5.7. Slightly different reconstructions of the trajectory are obtained due to the stochastic nature of the LMedS algorithm used to estimate the essential matrix. Although locally the visually estimated displacements are correct we see that the global estimate is not that accurate; the estimated endpoint of the trajectory is quite far from the actual endpoint.

Correcting visual odometry. The results shown in the previous section showed that the pose at the end of the trajectory is quite far from the true end pose due to accumulation of small errors. Wheel odometry suffers from the same phenomenon. Unlike wheel odometry, visual odometry offers the appealing possibility of recognizing a previously visited place on the basis of visual comparison. Although we applied our method to estimate the displacement only between subsequent images in the sequence, the visual method can be used to estimate the motion between any pair of images provided that there is sufficient visual overlap between the ground plane images and sufficient correspondences can be found in the panoramic images. The advantage of the visual method now becomes obvious. If we also estimate the motion between the last pose and the initial pose by means of our method, the trajectory can be "corrected".

As a demonstration, we used our relative pose estimation method to estimate the displacement between the last but one and the first image. We then get an estimate of the last pose, say C_N^*, which is linked directly to the initial pose. We also obtained an estimated of the last pose, say C_N, by composing the relative pose relationships leading up to the last pose. A simple method to correct the trajectory is to back-propagate the difference between the two estimates of the final pose proportionally as

$$C_i \leftarrow C_i - \left(1 - \frac{N-i}{N}\right)(C_N^* - C_N), \tag{5.32}$$



(a) estimated trajectory 1

(b) estimated trajectory 2

(c) estimated trajectory 3

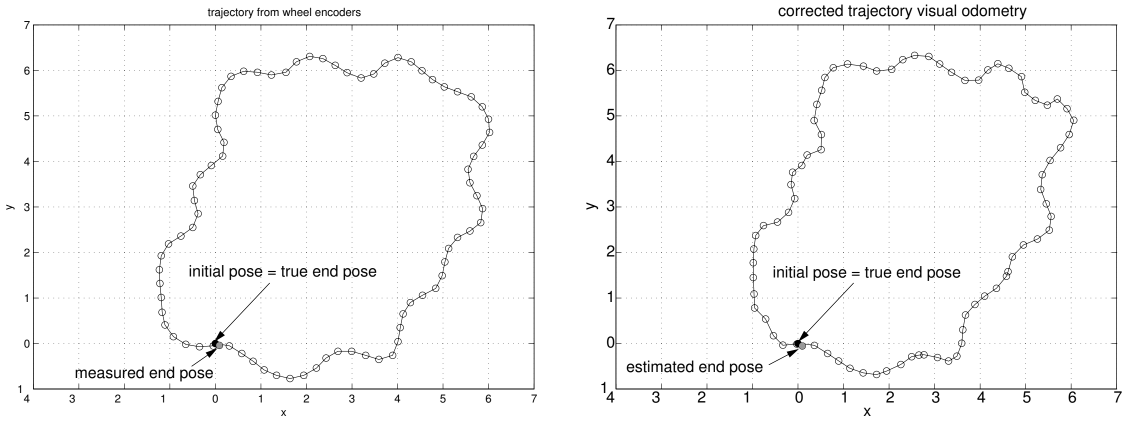
(d) estimated trajectory 4

Figure 5.7: Four examples of visually estimated trajectories. The start position (0,0) is marked by a black dot. The true trajectory is a closed-loop trajectory that ends exactly at the initial position. The small differences in the shown estimated trajectories are due to the stochastic nature of the Least of Median Squares estimator used to estimate the essential matrix relating subsequent images.

where C_i denotes the i-th pose and $(C_N^* - C_N)$ is the error at the last pose. The results of this simple correction technique are shown in figure 5.8b. The improvement is evident; although still not perfect, the trajectory resembles the trajectory measured by the wheel encoders closely.

Comparing the trajectory measured by the wheel encoders in figure 5.8a and the visually estimated trajectory shown in figure 5.8b shows that baseline lengths between the 7-th up to the 12-th image (in the area $1 < x < 3, -1 < y < 0$) are often underestimated. The corresponding images are shown in figure 5.9. These images contain relatively little texture so that the registration of these images as a function of the baseline scale is uncertain.

Further improvements of the estimated trajectory are possible if the uncertainty associ-



(a) wheel odometry

(b) corrected visual odometry

Figure 5.8: a) Path traversed by the robot according to wheel odometry. b) Path traversed by the robot according to visual odometry. The path was corrected by also estimating the pose estimate relating the last but one and the first image, and back-propagating the error.

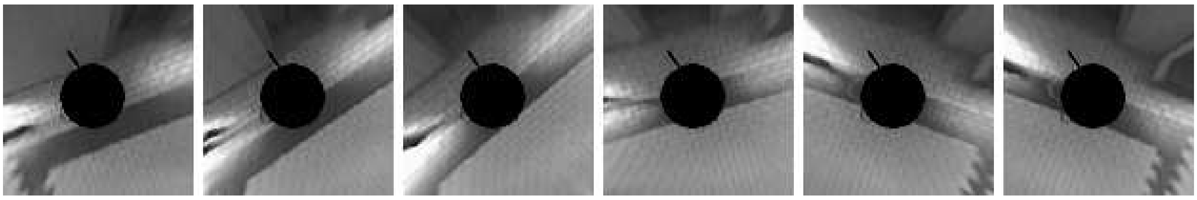


Figure 5.9: Image 7 through image 12. For these images, the baseline length is consistently underestimated. This is likely to be caused by excessive specular reflections and the absence of distinguishing texture on the ground plane.

ated with each relative pose estimate would be taken into account. For instance, instead of a simple proportional back-propagation of the error at the end-pose, a more principled Kalman-smoothing procedure as proposed in [88] could be used. The trajectory could even further be improved by estimating displacements and associated uncertainty for all image pairs. A globally consistent pose estimation technique, such as presented in [53] could then be applied.

5.5 Discussion and Conclusion

In this chapter we presented a method for estimating the translation and rotation between two subsequent poses of a moving robot from images taken with a panoramic catadioptric vision system. We have applied our method to obtain a form of visual odometry and have shown how a reconstruction of the past trajectory can be obtained. Our method operates in two stages and uses two different image representations derived from catadioptric images. In the first stage, point correspondences between cylindrical panoramic images are used to estimate the relative rotation and direction of translation between two camera viewpoints. In the second stage, perspective images of the ground plane are used to estimate the scale of the baseline by minimizing their intensity discrepancy as a function of the baseline length.

An issue of concern raised during experimentation was the extend to which "non-floor" pixels contribute to the error in the obtained baseline estimate. From our simulation experiments we conclude that the estimator is insensitive up to (at least) 57% "non-floor" pixels. Our visual odometry experiments suggest that the presence of sufficient texture on the ground plane appears to be the dominating factor.

Another issue of concern in using a two-stage procedure as we proposed is that the second stage relies on the accuracy of the first stage. If the rotation and direction of translation estimated in the first stage are wrong, the estimated baseline scale is also likely to be wrong. The visual odometry experiments we presented show that the rotation and direction of translation are accurately estimated in many cases. This positive result is in accordance with both theory [19] and experimental findings in other works employing omnidirectional or panoramic vision [28, 50, 68].

Although the visually estimated displacements are correct locally, when using visual odometry the estimated final pose can be far from the true final pose (which in our experiment is identical to the initial pose) due to error accumulation. Unlike wheel odometry, visual odometry is not "blind" in the sense that the return at a previously visited place can be detected. This opens up the opportunity to correct the past trajectory. We have used a simple method, which back-propagates the detected error over the past trajectory, to illustrate the principle. Our experimental results demonstrate that our two-stage pose estimation method can be used to obtain a good estimate of a past trajectory using visual information only.

Chapter 6

Summary, Conclusions and Future Work

This thesis discusses methods and techniques for map building and localization based on panoramic images captured during navigation in an indoor environment. In this chapter, we present general conclusions and indicate possible directions for future research.

Chapter 2. Chapter 2 introduces a vision sensor that observes the world through a hyperboloid shaped mirror. The vision sensor provides a panoramic field of view. We have presented a method to derive a mirror shape, height and size so that the resulting assembled vision sensor covers a specified vertical field of view. We presented a quick-and-easy calibration procedure, which estimates the camera focal length and the principal point. The calibration procedure assumes that other camera parameters are known, and that the mirror is positioned correctly with respect to the camera. The vision sensor measures light from a single point in space (the single effective viewpoint). As a result, images acquired by the sensor can be re-mapped to another surface. We have shown how a virtual cylindrical, spherical and planar perspective camera can be constructed. The usefulness of these virtual cameras is illustrated in subsequent chapters.

Chapter 3. Chapter 3 focuses on the problem of mapping and localization from panoramic images. The observation model — which can be regarded as a map — used for localization is estimated from a set of images labeled by their respective poses. We adopt an appearance-based approach that models the relation between images and robot poses directly. The advantage of such an approach over landmark-based approaches is that it does not rely on the extraction and matching of landmarks from images. Disadvantages of such an approach are that it may be sensitive to illumination changes and dynamic objects. We adopt a probabilistic approach for robot localization in which the robot maintains a belief function over the permissible poses in the workspace. The map for localization is represented by an observation model, a probability density function giving the likelihood of obtaining an observation given a robot pose. Prior to modeling, the dimensionality of the images is reduced by Principal Component Analysis (PCA). The observation model is then constructed from the resulting low-dimensional image representations, or PCA features, using a Parzen density estimator. We chose the Parzen model over alternative (parametric) models because it requires only estimation of a single parameter, the kernel bandwidth. We experimentally compared the performance of observation models instantiated using different PCA features. The performance of each model is optimized by minimizing the expected Bayesian localization error criterion [89]. We apply the same criterion for model selection. The performance of our observation model is evaluated under a situation in which the robot has to localize itself globally based on a single observation. Our experimental results show that the ordering of individual (1-D) PCA features according to an image reconstruction criterion is also optimal for localization. A tradeoff exists between localization accuracy and computational costs. Localization is generally more accurate if more features are used but is computationally more expensive if more features are used because localization involves computing distances in higher-dimensional spaces. Our experimental results show that an expected Bayesian localization error of 0.7 meters in an office environment of 17 × 17 meters can be achieved if the observation model is estimated from 300 training images, which are compressed to 15-D PCA features. Localization using such low-dimensional PCA features can be performed in real-time using current hardware technology.

Future directions. Although our experiments show that PCA yields good features for visual localization, better (linear) features may exist. Projection pursuit methods could be employed to search for features that optimize the expected Bayesian localization error. It can be expected that accurate localization then becomes possible using from very low-dimensional features. Recent work into this direction is presented in [102]. A drawback of PCA features is that they are affected by local occlusions in the image from which they are derived. Such occlusions frequently occur if a robot is operating in a dynamic environment where objects and people move around. A robust occlusion handling method is called for. Recent work addressing this issue is presented in [51]. Their method is based on the observation that the projection of an image into the PCA feature space can be regarded as an over-determined linear least squares system, which can be solved in a robust manner. In [42] the method is applied in a robot localization context. Occlusions often affect only a small part of a panoramic image. Based on this observation, [69] proposes a robust localization method in which cylindrical panoramic images are partitioned into overlapping image windows and localization is done on the basis of these "sub-views". Another localization approach based on image windows is presented in [109]. They propose an attention mechanism that automatically selects discriminating image windows, thereby gaining computational efficiency over other window-based approaches. Another issue that calls for a solution is the problem of orientation. The robot used in the experiments presented in chapter 3 maintains a constant orientation. One method to incorporate orientation in our localization method is to simply extend the database with rotated instantiations of the training images. A drawback of this method is that the number of training images required to represent the observation model expands substantially. Intuition tells us that we may do better; a rotated instantiation of an image does not contain information other than the information present in the original image. One approach that aims to address the issue is presented in [67]. They propose to shift cylindrical panoramic images so that their first harmonic in the Discrete Fourier Transform has a phase equal to zero. The transformation produces a single representative image for rotated instantiations of the same image. An issue of concern is the stability of such a method under varying illumination conditions and under small camera displacements.

Chapter 4. Chapter 4 focuses on the estimation of the 3-D structure from 2-D panoramic images acquired at known poses. We formalized the epipolar geometry for cylindrical panoramic cameras and show that epipolar curves for such cameras are sinusoids. We presented two methods to obtain a dense 3-D reconstruction from images. The first method uses an angular parameterization of epipolar curves in order to guide the search for image correspondences between a pair of images. We have shown that accurate reconstruction requires disparity estimates at sub-pixel precision. Our experimental results show that a depth map (and consequently the estimate of the 3-D structure) estimated from a single pair of images is noisy due to erroneous correspondences. Furthermore, depth in the direction of camera motion cannot be estimated accurately due to triangulation uncertainty. In order to address these issues we present a second method to obtain a 3-D reconstruction from multiple cylindrical panoramic images. We derived a parameterization of epipolar curves in terms of inverse depth, a quantity directly related to depth. This parameterization enables efficient search for correspondences across multiple images. Our experimental results demonstrate that depth maps of good quality can be obtained using our multi-image method.

Future directions. Depth maps estimated by the methods presented in chapter 4 contain erroneous depth values resulting from matching errors. An obvious extension that could handle such erroneous depth estimates, would be to employ a form of regularization as a post-processing step. We have shown how the estimated depth maps can be used to generate images that would be obtained from nearby camera poses. Using such image based warping in principle enables the generation of a large database of training images required by the appearance-based map building approach presented in chapter 3. Related work into this direction has been presented in [10] for range profiles.

Chapter 5. Chapter 5 focuses on the estimation of a trajectory from a sequence of camera images captured during navigation without using robot odometry. Batch methods proposed in literature require reliable a set of image correspondences across all images. Obtaining such a set of correspondences automatically is a fundamental problem in computer vision (correspondence problem). Sequential methods assume that images arrive sequentially so that tracking techniques can be employed to establish correspondences. Most existing sequential approaches attempt to integrate points tracked into a new image into a sparse 3-D model containing the 3-D positions of tracked features and previous camera poses. A drawback of such sequential approaches is that because of error accumulation the model may eventually become inconsistent. An issue particularly prominent if a single camera is used (instead of a stereo head) is that the scale of the 3-D model is subject to drift. We present a two-stage method that overcomes these issues by problem by exploiting prior knowledge that the camera undergoes a 2-D motion parallel to a planar floor. First, the relative pose between a pair of cylindrical panoramic images is estimated via the epipolar geometry, which can be estimated independently of the scene structure. Subsequently, the length of the baseline relating the two camera poses is obtained by minimizing the intensity discrepancy between two planar perspective images of the ground plane as a function of the baseline length. The cylindrical panoramic images and the planar perspective images of the ground plane are derived from the images acquired by the panoramic vision sensor described in chapter 2. Our experimental results show that the rotation and baseline direction relating consecutive cylindrical panoramic images can be estimated robustly from (noisy) tracked image features. The method used to estimate the baseline length from images of the ground plane implicitly assumes that only floor, and no extruding objects or specular reflections, are visible in the images. Simulation experiments show that the proposed method is robust against mild violations of this assumption. Good estimates of the baseline length can be achieved if up to 57% pixels in the images come from objects extruding from the ground plane provided that the plane is sufficiently textured. Simulation experiments furthermore indicate that inclusion of pixels coming from extruding object may provide some robustness against the corruption of the baseline length estimate caused by specular reflections. We attribute this phenomenon to the (often large) image gradient information present at the boundaries between floor and extruding objects which would otherwise have been ignored; the method used to minimize the intensity discrepancy between two images of the ground plane crucially relies on the presence of image gradient information. An estimate of the trajectory can be obtained by concatenating the relative pose estimates relating consecutive images. Like wheel odometry, such visual odometry suffers from the accumulation of errors. We have demonstrated that the estimated trajectory can be corrected considerably if we also estimate the relative pose relation between the first and the last image in the sequence (which are obtained at nearby poses in our experiment) using our visual method and back-propagate the error at the last pose.

Future directions. In chapter 5, we have based the estimate of the trajectory on visual estimates of relative pose relationships between consecutive images only. Using only consecutive images enables visual tracking of salient image features. Corresponding image features are required in order to infer a relative pose relationship between two images. The trajectory estimate could be improved if the uncertainty of each estimated relative pose relationship would also be estimated and taken into account in reconstructing the trajectory. Further improvements of the trajectory estimate are possible if the relative pose relationship between every possible pair of images would be estimated using our visual method. Globally consistent alignment methods [53], which aim to find a set of poses consistent with all relative pose estimates, can then be employed. The main challenge of such an extension would be establishing correspondences between pairs of images automatically because simple feature tracking techniques can no longer be used.

Concluding remarks. Current generation mobile service robots are not truly autonomous. In order to perform their navigational tasks, they have to be equipped with a pre-specified map of their workspace and they often rely on the presence of uniquely identifiable artificial landmarks. In this thesis, we have presented visual methods for mapping and localization, 3-D structure estimation and trajectory estimation from a panoramic vision sensor. The research presented in this thesis is a modest contribution to a next generation of mobile service robots; affordable seeing mobile robots which are capable of building and maintaining their own maps while performing their service tasks in everyday indoor environments.

Appendix A

Principal Component Analysis

If images are regarded as D-dimensional vectors, Principal Component Analysis calculates the eigenvectors and their associated eigenvalues of the covariance matrix of a set of images. The eigenvectors span a new D-dimensional orthogonal basis in which images can be represented. By using only a few $K \ll D$ eigenvectors with the largest corresponding eigenvalues, the dimensionality of the images is reduced with minimal loss of variance present in the original images. The K-dimensional subspace in called the eigenspace. Although perfect reconstruction of an image from its eigenspace representation generally requires that $K = D$, low-dimensional eigenspace image representations suffice for visual recognition [61].

Let $\mathbf{z}$ denote a D dimensional column vector which is formed by a chosen ordering of the pixels in an image. If a set $\{\mathbf{z}_1, \ldots, \mathbf{z}_L\}$ of images is available, we first subtract the mean image from each image in the set. The mean image is calculated as

$$\bar{\mathbf{z}} = \frac{1}{L} \sum_{i=1}^{L} \mathbf{z}_i. \tag{A.1}$$

The mean subtracted images are subsequently stacked to form a $D \times L$ image matrix

$$\mathbf{Z} = [\mathbf{z}_1 - \bar{\mathbf{z}}, \ldots, \mathbf{z} - \bar{\mathbf{z}}]. \tag{A.2}$$

Subtraction of the mean ensures that the eigenvector with the largest eigenvalue represents the dimension in eigenspace for which the variance of the images is maximal. The $D \times D$ covariance matrix $\mathbf{\Sigma}$ of $\mathbf{Z}$ is defined as

$$\mathbf{\Sigma} = \mathbf{Z}\mathbf{Z}^T. \tag{A.3}$$

The Eigenvectors $\mathbf{b}_i$ and corresponding eigenvalues λ_i of $\mathbf{\Sigma}$ are determined by solving the well-known eigenstructure decomposition problem

$$\lambda_i \mathbf{b}_i = \mathbf{\Sigma} \mathbf{b}_i, \tag{A.4}$$

which can be expressed in matrix form as

$$\mathbf{\Lambda} = \mathbf{\Phi}^T \mathbf{\Sigma} \mathbf{\Phi}, \tag{A.5}$$

where $\mathbf{\Lambda}$ is a $D \times D$ diagonal matrix containing the eigenvalues λ_i and $\mathbf{\Phi}$ is a $D \times D$ matrix whose columns constitute the Eigenvectors $\mathbf{b}_i$. After sorting the Eigenvectors in descending order of corresponding eigenvalues, the first $K \ll D$ Eigenvectors are used to project an image vector $\mathbf{z}$ into the K-dimensional eigenspace

$$\mathbf{y} = \mathbf{\Phi}_K^T [\mathbf{z} - \bar{\mathbf{z}}]. \tag{A.6}$$

The calculation of the Eigenvectors of a large matrix is computationally expensive. Several methods have been developed to compute the Eigenvectors efficiently. Singular value decomposition (SVD) [73] is numerically the most accurate way to compute the Eigenvectors. Singular value decomposition can be directly applied to decompose the covariance matrix $\mathbf{\Sigma}$ as $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{V}$ is a $D \times D$ orthonormal matrix whose columns are the Eigenvectors $\mathbf{b}_i$. Using SVD, $\mathbf{\Lambda}$ contains the eigenvalues λ_i in increasing order. Often, the number of training vectors is much smaller than the dimensionality of the training vectors, i.e. $L \ll D$. In [49] it is shown that the Eigenvectors and eigenvalues of $\mathbf{\Sigma}$ can also be derived via an SVD decomposition of the implicit covariance matrix defined by $\tilde{\mathbf{\Sigma}} = \mathbf{Z}^T\mathbf{Z}$. If $L < D$ the implicit covariance matrix is smaller than the covariance matrix and hence its SVD decomposition can be calculated faster. The Eigenvectors and eigenvalues of the covariance matrix can be derived as

$$\begin{aligned} \lambda_i &= \tilde{\lambda}_i \\ \mathbf{e}_i &= \tilde{\lambda}_i^{-1/2} \mathbf{Z} \tilde{\mathbf{e}}_i, \end{aligned} \tag{A.7}$$

where λ_i and $\mathbf{e}_i$ denote the i-th eigenvalue and eigenvector of $\mathbf{\Sigma}$, and $\tilde{\lambda}_i$ and $\tilde{\mathbf{e}}_i$ denote the corresponding eigenvalue and eigenvector of $\tilde{\mathbf{\Sigma}}$. We adopt this method.

We would like to inform the reader that more efficient methods have been devised for sets of images which are in-plane rotated realizations of the same image. Such in-plane rotated realizations of a same image can be obtained by our catadioptric vision system is rotated about the mirror axis of symmetry. In [99] it is shown that the Eigenvectors of a set of such rotated images which are obtained from a single image are the basis vectors for the discrete cosine transform of the original image in polar coordinates. As a result, the Eigenvectors of the rotated images can be obtained efficiently. A similar method is developed in [7]. In [44] these methods are extended to the case of rotated images obtained from multiple viewpoints.

Appendix B

Belief Function Representations

The key technical difficulty in implementing the recursive localization approach presented in section 3.3 is to maintain an accurate representation of the belief function and to do so efficiently. In this section we outline some important approaches that have been proposed in literature.

Kalman filtering. A traditional approach to recursive state estimation is Kalman filtering. The Kalman filter represents posteriors by a Gaussian distributions. A Gaussian distribution is completely characterized by its mean and covariance matrix. The Gaussian distribution is closed under linear transformations. In a Kalman filter this property is exploited to derive a posterior analytically under this assumption. The Gaussian distribution is however not closed under convolution with an arbitrary motion model or multiplication with an arbitrary observation model. These happen to be the main steps in probabilistic robot localization. This means that even if at some point in time the belief is Gaussian, the posterior derived from a new measurement will generally not be Gaussian. This problem can be overcome partially if the sensor model and motion model are linearized (extended Kalman filtering, unscented Kalman filtering). The extended Kalman filter maintains a Gaussian belief which approximates the true posterior. The quality of the approximation depends on the uncertainty of the current state estimate. Poor approximations generally result when the true posterior is multi-modal. This typically occurs when the robot is very uncertain about its pose. In this case, only one of these peaks will be retained by the Kalman filter. The use of Kalman filtering is therefore restricted to *pose tracking*, where the robot initially knows its location. As a straightforward extension of the Kalman filter, multi-hypothesis Kalman filtering has been proposed, which use a mixture of Gaussians to represent multiple hypotheses concerning the robot pose [40].

Grid-based methods. A popular alternative are *grid-based* methods. In grid based methods, densities are modeled as piece-wise constant functions (histograms). The workspace of the robot is discretized into cells. The value of each cell reflects the likelihood that the robot is located somewhere in the area covered by the cell. Grid based methods can accurately represent multi-modal distributions. The localization accuracy that can be obtained is limited by the grid resolution. Another problem of the naive grid based method is that the memory and computation requirements grow proportional in the number of cells. To overcome these issues, several enhancements such as octree decompositions and geometric hashing methods have been proposed in literature [5].

Particle-based methods. Particle-based methods are quickly gaining popularity (see [14] for an overview of the state of the art). Particle based methods approximate posterior densities by a weighted set of m samples (called particles). The weights are called importance factors. The basic algorithm is as follows. The initial density is represented by a uniform sample of fixed size. When the robot has moved, a new set of m samples is drawn according to the importance factors. Samples with a larger importance factor are thus more likely to be drawn. For each such drawn sample, a successor location is guessed according to the motion model. If a new observation arrives, a new importance factor is calculated for each sample according to the observation model. Finally, the importance factors are normalized such that their sum equals unity.

The particle based algorithms are popular because they are relatively easy to implement, yet can faithfully represent arbitrary densities provided that enough particles are used. There are many variants of the basic algorithm. Modifications have been proposed to adapt the size of the sample set dynamically. The rationale is that fewer particles are needed to accurately approximate a density which has a few narrow peaks. One problem of the basic algorithm is that particles are drawn from the prior $p(x)$. This may fail to produce enough particles in the overlapping region between the prior and the likelihood $p(o|x)$. This may cause the posterior may be poorly represented. As a result, a robot that is well localized may get lost because the posterior is poorly represented. One way to recover from such situations is to insert some random samples. More advanced methods have been proposed which do not just sample from the prior, but instead aim to optimally sample from the posterior [105]. These methods have been shown to be able to recover from situations where a prior is peaked at the wrong location (the so-called *kidnapped robot problem*).

Bibliography

[1] CO. LTD. Accowle. WWW: `http://www.pluto.dti.ne.jp/~accowle1/`.

[2] R. Benosman and S.B. Kang, editors. *Panoramic Vision: sensors, theory and applications*. Monographs in Computer Science. Springer-Verlag, New York, first edition, 2001.

[3] S. Birchfield. KLT: An implementation of the kanade-lucas-tomasi feature tracker. Available at: `http://vision.stanford.edu/~birch/klt/`.

[4] S. Bogner. Introduction to panoramic imaging. In *Proc. IEEE SMC Conf.*, pages 3100–3106, 1995.

[5] W. Burgard, D. Fox, and D. Hennig. Fast grid-based position tracking for mobile robots. In *Proc. of the German Conference on Artificial Intelligence (KI)*, number 1303 in Lecture Notes in Computer Science, Germany, 1997. Springer-Verlag.

[6] J.S. Chahl and M.V. Srinivasan. Reflective surfaces for panoramic imaging. *Applied Optics*, 36(31):8276–8285, November 1997.

[7] C.Y. Chang, A.A. Maciejewski, and Balakrishnan V. Fast eigenspace decomposition of correlated images. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 7–12, Victoria, B.C., Canada, October 1998.

[8] R.T. Collins. A space-sweep approach to true multi-image matching. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 358–363, 1996.

[9] I.J Cox. Blanche – an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Trans. on Robotics and Automation*, 7(2):193–204, 1991.

[10] J.L. Crowley, F. Wallner, and B. Schiele. Position estimation using principal components of range data. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3121–3128, Leuven, Belgium, May 1998.

[11] R. Cutler, Y. Rui, A. Gupta, J.J. Cadiz, I. Tashev, L. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg. Distributed meetings: A meeting capture and broadcasting system. In *Proc. ACM Conf. on Multimedia*, Juan Les Pins, France, December 2002.

[12] K. Deguchi. A direct interpretation of dynamic images and camera motion for vision guided robotics. In *Proc. IEEE/SICE/RSJ Int. Conf. on Multisensor Fusion*, pages 313–320, 1996.

[13] P. Doubek and T. Svoboda. Reliable 3D reconstruction from a few catadioptric images. In Proc. IEEE 3rd Workshop on Omnidirectional Vision [75].

[14] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[15] R.P.W. Duin. Smoothing parameters for parzen estimators. *IEEE Trans. on Computers*, pages 1165–1179, 1976.

[16] J. Fabrizio, J.P. Tarel, and R. Benosman. Calibration of panoramic catadioptric sensors made easier. In Proc. IEEE 3rd Workshop on Omnidirectional Vision [75].

[17] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, Massachusetts, 1993.

[18] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. Technical Report 856, INRIA, June 1988.

[19] C. Fermüller and Y. Aloimonos. Algorithm-independent stability analysis of structure from motion. Technical Report CS-TR-3691, Computer Vision Laboratory, University of Maryland, 1996.

[20] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA, 2nd edition, 1990.

[21] P. Fornland. Dominant plane detection for uncalibrated binocular vision. In *Proc. Int. Symp. on Intelligent Robotic Systems (ISIRS)*, Bangalore, India, January 1998.

[22] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. In *Proc. 6th National Conf. on Artificial Intelligence; Proc. 11th Conf. on Innovative Applications of Artificial Intelligence*, pages 343–349, Menlo Park, Cal., July 18–22 1999. AAAI/MIT Press.

[23] E. A. Fox. *Mechanics*. Harper and Row, 1967. (see also WWW: `http://www.madscitech.org/theorist/systems5/systems5.html`).

[24] Fullview. WWW: `http://www.fullview.com`.

[25] J. Gaspar, J. Santos-Victor, and J. Sentieiro. Ground plane obstacle detection with a stereo vision system. In *Int. Workshop on Intelligent Robotic Systems*, Grenoble, France, July 1994.

[26] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omni-directional camera. *IEEE Trans. on Robotics and Automation*, 16(6):890–898, December 2000.

[27] C. Geyer and K. Daniilidis. Catadioptric camera calibration. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 398–404, Kerkyra, Greece, September 1999.

[28] J. Gluckman and S. K. Nayar. Ego-motion and omnidirectional cameras. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 999–1005, Bombay, India, 1998.

[29] J. Gluckman, S.K. Nayar, and K.J. Thoresz. Real-time omnidirectional and panoramic stereo. In *Proc. Image Understanding Workshop*, pages 299–303, 1998.

[30] R.I. Hartley. Estimation of relative camera positions from uncalibrated cameras. In *Proc. 2nd European Conf. on Computer Vision*, number 588 in Lecture Notes in Computer Science, pages 579–587. Springer-Verlag, May 1992.

[31] R.I. Hartley. In defence of the 8-point algorithm. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1064–1070, June 1995.

[32] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[33] J. Heikkilä and O. Silvén. A four-step camera calibration procedure with implicit image correction. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1112, Puerto Rico, June 1997.

[34] R.A. Hicks and R. Bajcsy. Reflective surfaces as computational sensors. *IVC*, 19(11):773–777, September 2001.

[35] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. Riseman. Image-based homing. *IEEE Control Systems Magazine*, 12(1):38–45, 1992.

[36] B.K.P. Horn. Recovering baseline and orientation from essential matrix, January 1990. WWW: `http://www.ai.mit.edu/people/bkph/publications.html`.

[37] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 29(1):5–28, 1998.

[38] H. Ishiguro. Development of low-cost and compact omnidirectional vision sensors. In Benosman and Kang [2], chapter 3, pages 23–38.

[39] M. Jägersand. Image based view synthesis of articulated agents. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1047–1053, 1997.

[40] P. Jensfelt and S. Kristensen. Active global localisation for a mobile robot using multiple hypothesis. In *Proc. of the Workshop on Reasoning with Uncertainty in Robot Navigation (IJCAI'99)*, pages 13–22, Stockholm, Sweden, August 1999.

[41] M. Jogan and A. Leonardis. Panoramic eigenimages for spatial localisation. In *Proc. Int. Conf. on Computer Analysis of Images and Patterns (CAIP)*, number 1689 in Lecture Notes in Computer Science, pages 558–567. Springer Verlag, 1999.

[42] M. Jogan and A. Leonardis. Robust localization using panoramic view-based recognition. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*, pages 136–139, Barcelona, 2000.

[43] M. Jogan and A. Leonardis. Robust localization using the eigenspace of spinning-images. In Proc. IEEE 1st Workshop on Omnidirectional Vision [74], pages 37–44.

[44] M. Jogan and A. Leonardis. Parametric eigenspace representations of panoramic images. In *Proc. Int. Conf. on Advanced Robotics (ICAR)*, pages 31–36, Budapest, 2001. Workshop on Omnidirectional Vision Applied to Robotic Orientation and Nondestructive Testing (NDT).

[45] T. Kanade, H. Kano, S. Kimura, A. Yoshida, and K. Oda. Development of a video-rate stereo machine. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 95–100, Pittsburgh PA., 1995.

[46] S.B. Kang. Catadioptric self-calibration. In *CVPR*, Hilton Head Island, SC, 2000.

[47] K. Konolige. Small vision systems: Hardware and implementation. In *Proc. 8th Int. Symposium on Robotics Research*, Hayama, Japan, 1997.

[48] K. Konolige and K. Chou. Markov localization using correlation. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 1154–1159, 1999.

[49] V. Kumar and H. Murakami. Efficient calculation of primary images from a set of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 4(5):511–515, September 1982.

[50] J.W. Lee, S. You, and U. Neumann. Large motion estimation for omnidirectional vision. In Proc. IEEE 1st Workshop on Omnidirectional Vision [74], pages 161–168.

[51] A Leonardis, H. Bischof, and R. Ebensberger. Robust recognition using eigenimages. Technical Report PRIP-TR-47, PRIP, Viena University of Technology, June 1997.

[52] M.I.A. Lourakis and S.C. Orphanoudakis. Visual detection of obstacles assuming a locally planar ground. In *Proc. of the Asian Conf. on Computer Vision*, volume 2, pages 527–534, Hong Kong, Januari 1998.

[53] F. Lu and E. Millios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[54] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. DARPA Image Understanding Workshop*, pages 121–130, 1981.

[55] S. Maeda, Y. Kuno, and Y. Shirai. Active navigation vision based on eigenspace analysis. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1018–1023, 1997.

[56] J. Martin and J.L. Crowley. Comparison of correlation techniques. In *Proc. Symp. on Robotic Systems (SIRS)*, Karlsruhe, Germany, 1995.

[57] L. Matthies. Stereo vision for panetary rovers: Stochastic modeling for near real-time implementation. *Int. Journal of Computer Vision*, 8(1):71–91, 1992.

[58] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *Proc. of SIGGRAPH 95*, pages 39–46, Los Angeles, CA, 1995.

[59] Immersive Media. WWW: `http://www.immersivemedia.com`.

[60] H.P. Moravec. Robot spatial perception by stereoscopic vision and 3-D evidence grids. Technical Report CMU-RI-TR-96-34, The Robotics Institute, Carnegie Mellon Univ., Pittsburgh, Pennsylvania, September 1996.

[61] H. Murase and S.K. Nayar. Visual learning and recognition of 3-D objects from appearance. *Int. Journal of Computer Vision*, 14:5–24, 1995.

[62] S. Nayar, S. Nene S, and H. Murase. Subspace methods for robot vision. Technical Report TR CUCS-06-95 CS, Dept. of Computer Science, Columbia Univ., New York, February 1995.

[63] S.K. Nayar and S. Baker. A theory of catadioptric image formation. Technical Report CUCS-015-097, Dept. of Computer Science, Columbia Univ., 1997.

[64] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 15(4):353–363, April 1993.

[65] M. Ollis, H. Herman, and S. Singh. Analysis and design of panoramic stereo vision using equi-angular pixel cameras. Technical Report CMU-RI-TR-99-04, The Robotics Institute, Carnegie Mellon University, Pittsburgh, USA, 1999.

[66] S. Oore, G.E. Hinton, and G. Dudek. A mobile robot that learns its place. *Neural Computation*, 9:683–699, 1997.

[67] T. Pajdla. Robot localization using panoramic images. In *Proc. Computer Vision Winter Workshop*, pages 1–12, Rastenfeld, Austria, 1999.

[68] T. Pajdla, T. Svoboda, and V. Hlaváč. Epipolar geometry of central panoramic cameras. In Benosman and Kang [2], chapter 5, pages 73–102.

[69] L. Paletta, S. Frintrop, and J. Hertzberg. Robust localization using context in omnidirectional imaging. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2072–2077, Seoul, Korea, May 2001.

[70] E. Parzen. On the estimation of a probability density function and the mode. *Ann. Math. Statist.*, 33:1065–1076, 1962.

[71] F. Pourraz and J.L. Crowley. Continuity properties of the appearance manifold for mobile robot estimation. In *Proc. Symp. on Robotic Systems (SIRS)*, Edinburgh, 1998.

[72] POV-Ray Team. Persistence of vision ray tracer, 1999. Available at WWW: `http://www.povray.org`.

[73] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++.* Cambridge University Press, 2nd edition, 2002.

[74] *Proc. IEEE 1st Workshop on Omnidirectional Vision*, Hilton Head, South Carolina, June 2000.

[75] *Proc. IEEE 3rd Workshop on Omnidirectional Vision 2002*, Copenhagen, Denmark, June 2002.

[76] D.W. Rees. Panoramic television viewing system. U.S. Patent No. 3,505,465, April 1970.

[77] B.M. Ter Haar Romeny. Applications of scale-space theory. In J. Sporring et al., editor, *Gaussian Scale-Space*, volume 8 of *Computational Imaging and Vision*, chapter 1. Kluwer Academic Press, 1997.

[78] P.J. Rousseeuw. *Robust Regression and Outlier Detection.* Wiley, New York, 1987.

[79] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1067–1073, 1997.

[80] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[81] H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *Int. Journal of Computer Vision*, 36(2), 2000.

[82] T. Svoboda. Evaluation, transformation, and parameterization of epipolar conics. Technical Report CTU-CMP-2000-11, Center of Machine Perception, Czech Technical Univ., 2000.

[83] T. Svoboda and T. Pajdla. Matching in catadioptric images with appropriate windows and outliers removal. In *Proc. Int. Conf. on Computer Analysis of Images and Patterns (CAIP)*, pages 733–740, Berlin, Germany, September 2001.

[84] T. Svoboda, T. Pajdla, and V. Hlaváč. Central panoramic cameras: Design and geometry. In *Proc. Computer Vision Winter Workshop*, Gozd Martuljek, Slovenia, 1998.

[85] T. Svoboda, T. Pajdla, and V. Hlaváč. Epipolar geometry for panoramic cameras. In *Proc. European Conf. on Computer Vision*, volume 1406 of *Lecture Notes in Computer Science*, pages 218–232, Freiburg, Germany, 1998. Springer.

[86] T. Svoboda, T. Pajdla, and V. Hlaváč. Motion estimation using central panoramic cameras. In *Proc. IEEE Conf. on Intelligent Vehicles*, pages 335–340, Stuttgard, Germany, 1998.

[87] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, March 1996.

[88] S.H.G. ten Hagen and B.J.A. Kröse. Trajectory reconstruction for self-localization and map building. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1796–1801, Washington D.C., USA, May 2002.

[89] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1):41–76, 1998.

[90] S. Thrun. Finding landmarks for mobile robot navigation. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 958–963. IEEE Press, 1998.

[91] S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93–109, 2000.

[92] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998.

[93] P.H.S. Torr, A.W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *Int. Journal of Computer Vision*, 32(1):27–44, August 1999.

[94] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. Journal of Computer Vision*, 24(3):271–300, 1997.

[95] B. Triggs. Autocalibration from planar scenes. In *Proc. European Conf. on Computer Vision*, pages 89–105, 1998.

[96] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision.* Prentice Hall, Upper Saddle River NJ, 1998.

[97] R.Y. et al. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 364–74, Miami Beach, FL, USA, June 1986.

[98] R.Y. et al Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Trans. on Robotics and Automation*, 3(4):323–344, August 1987.

[99] M. Uenohara and T. Kanade. Optimal approximation of uniformly rotated images: Relationship between karhunen-loeve expansion and discrete cosine transform. *IEEE Trans. on Image Processing*, 7(1):116–119, January 1998.

[100] W. van der Mark. Autonome voertuig geleiding d.m.v. stereovisie. Masters thesis, University of Amsterdam, 2000. (in Dutch).

[101] Viewplus. WWW: `http://www.viewplus.co.jp`.

[102] N. Vlassis, R. Bunschoten, and B. Kröse. Learning task-relevant features from robot data. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 499–504, Seoul, Korea, May 2001.

[103] N. Vlassis and B. Kröse. Mixture conditional density estimation with the EM algorithm. In *Int. Conf. on Artificial Neural Networks (ICCAN)*, pages 821–825, 1999.

[104] N. Vlassis and B. Kröse. Robot environment modeling via principal component regression. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 667–682, 1999.

[105] N. Vlassis, B. Terwijn, and B. Kröse. Auxiliary particle filter robot localization from high-dimensional sensor observations. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 7–12, Washington, D.C., May 2002.

[106] S. Wei. Stereo matching of catadioptric panoramic images. Technical Report CTU-CMP-2000-08, Center for Machine Perception, Czech Technical Univ., 2000.

[107] R. Wilson. Tsai camera calibration software. Available at WWW: `http://www.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html`.

[108] N. Winters and J. Santos-Victor. Mobile robot navigation using omni-directional vision. In *Proc. 3rd Irish Machine Vision and Image Processing Conf.*, Dublin, Ireland, September 1999.

[109] N. Winters and J. Santos-Victor. Information sampling for vision-based robot navigation. *Robotics and Autonomous Systems*, (41):145–159, 2002.

[110] G. Wolberg. *Digital Image Warping.* IEEE Computer Society Press, 1990.

[111] Y. Yagi and S. Kawato. Panoramic scene analysis with conic projection. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 1990.

[112] Y. Yagi, S. Kawato, and S. Tsuji. Real-time omnidirectional image sensor (copis) for vision-guided navigation. *IEEE Trans. on Robotics and Automation*, 10(1):11–21, February 1994.

[113] Y. Yagi, K. Shouya, and M. Yachida. Environmental map generation and egomotion estimation in a dynamic environment for an omnidirectional image sensor. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3493–3498, San Francisco, 2000.

[114] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. Technical Report 2676, INRIA, October 1995.

[115] J.Y. Zheng and S. Tsuji. Color-based panoramic representation of outdoor environment for a mobile robot. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*, volume 2, pages 801–803, 1988.

[116] G. Ziegler. *Region-based analysis and coding of stereoscopic video.* PhD thesis, Technical University Delft, 1997.

Summary

This thesis presents and discusses methods and algorithms that enable a mobile robot to learn an internal representation of its workspace. An internal representation is required to correct a pose estimate, which grows more uncertain when the robot navigates, based on new observations. We consider an application in which the robot is equipped with a panoramic vision sensor providing a 360° field of view.

First, we describe the panoramic vision sensor. The sensor consists of a conventional camera, which is aimed at a convex mirror. Although the camera is mounted below the mirror, the image observed by the camera can be regarded as though it is obtained from a single, fixed viewpoint residing inside the mirror. We use this property to construct virtual cameras. These virtual cameras share the same viewpoint, but yield a projection onto a different surface (for instance a cylinder). Throughout the thesis we show that images from such virtual cameras lend themselves better for certain tasks than the panoramic images from which they are derived. We also propose a quick and simple calibration method for the panoramic vision sensor.

Next, we discuss a method to estimate an internal model for localization (pose estimation) from a set of training samples (images and their associated pose). Our approach fits within a Bayesian framework for localization (Markov localization) where the estimate of the robot's pose at any time is represented as a probability density function. In order to adopt the Bayesian approach it is essential that low-dimensional features be extracted from the images. In our application, we use global features obtained by performing a principal component analysis (PCA) on the set of training images (appearance-based approach). We maximize the performance of the model by minimizing the expected Bayesian localization error. The same criterion is applied to determine which and how many PCA features are needed for global localization from a single observation.

Next, we focus on the problem of obtaining a 3-D reconstruction from a collection of images using stereo vision. We show that for (virtual) cylindrical panoramic images, the epipolar curves (along which image correspondences must be sought) are sinusoids. The first stereo vision method we present uses an angular parameterization of these epipolar curves. The chosen parameterization is suitable to obtain a reconstruction from a pair of images. A disadvantage when only using a pair of images is that the resulting depth image contains many errors. Furthermore, it is impossible to obtain reliable depth estimates in the direction of camera displacement. We propose a second method that addresses these issues. We derive a parameterization of epipolar curves in terms of a quantity directly related to depth. This parameterization enables us to combine depth estimates obtained from different image pairs yielding a more reliable depth estimate.

The previously discussed methods assume that the input images are obtained at known camera poses. As a final contribution, we present a two-stage method to automatically estimate relative camera poses from pairs of images. Our method assumes that the camera motion is parallel to the ground plane. In the first stage, the relative orientation and direction of translation are estimated from image correspondences (established across virtual cylindrical panoramic images). In the second stage, the length of the translation vector is estimated based on images of the ground plane (obtained by a virtual planar perspective camera aimed straight at the ground plane). In practice, the first stage yields a robust estimate of the relative orientation and direction of translation. Simulation experiments show that the second stage gives a robust estimate of the length of the translation vector in the presence of specular reflections and objects extruding from the ground plane. We apply the proposed method to reconstruct a past trajectory from a sequence of images acquired during navigation. The relative pose estimates from successive images (neighboring in time) can be combined in a simple manner to obtain a reconstruction of the past trajectory. Applied in this fashion, a form of visual odometry is obtained. Like wheel odometry, this approach is subject to accumulation of (estimation) errors. We demonstrate that the effects of error propagation can be compensated for by considering images that are neighboring in space, but not necessarily in time.

The value of the methods and algorithms proposed in this thesis are endorsed by experiments performed on real-world images.

Samenvatting*

Dit proefschrift behandelt en bediscussieert methoden en algoritmen welke een mobiele robot in staat stellen om op basis van camera waarnemingen een interne representatie van zijn omgeving te vormen. Een interne representatie is nodig om een schatting van de pose van de robot, die onzekerder wanneer de robot rondrijdt, te kunnen corrigeren op basis van nieuwe waarnemingen. We beschouwen een robot die is uitgerust met een panoramisch camera systeem waarmee 360 graden rondom waargenomen wordt.

Eerst volgt een beschrijving van het panoramische camera systeem. Het systeem bestaat uit een gangbare camera die op een gebolde spiegel is gericht. Hoewel de camera onder de spiegel is gemonteerd, kan een beeld dat door de camera is vergaard beschouwd worden alsof het is waargenomen vanuit een vastliggend punt in de spiegel. We maken van deze eigenschap gebruik om virtuele camera's te construeren. Deze virtuele camera's geven een projectie op een ander oppervlak (bijvoorbeeld een cilinder). In de loop van het proefschrift laten we zien dat beelden van dergelijke virtuele camera's zich voor bepaalde taken beter lenen dan de panoramische beelden waarvan zij zijn afgeleid. We stellen tevens een kalibratie methode voor waarmee het panoramische camera systeem snel en eenvoudig gekalibreerd kan worden.

Vervolgens behandelen we een methode welke op basis van een set leervoorbeelden (bestaande uit beelden en hun geassocieerde camera pose) een intern model van de omgeving schat dat gebruikt kan worden voor lokalisatie (pose schatten). De gekozen benadering past binnen een Bayesiaans raamwerk voor lokalisatie (Markov lokalisatie) waarin de schatting van de pose van de robot op ieder moment wordt gerepresenteerd door een kansdichtheidsfunctie. Om de Bayesiaanse benadering te kunnen volgen is het essentieel dat laagdimensionale kenmerken uit de beelden worden geëxtraheerd. In onze toepassing kiezen we daarbij voor globale beeldkenmerken welke middels Principale Componenten Analyse (PCA) worden verkregen uit de set van leerbeelden (appearance-based aanpak). De prestatie van het model maximaliseren we door de verwachte lokalisatiefout te minimaliseren. Datzelfde criterium gebruiken we om te bepalen welke en hoeveel PCA kenmerken nodig zijn voor globale lokalisatie op basis van één observatie.

Daarna richten we ons op het reconstrueren van de 3-D omgeving uit een set van beelden middels stereovisie. We laten zien dat in het geval van (virtuele) cilindrische panorama

*Summary in Dutch.

camera's de epipolaire lijnen (waarlangs naar puntcorrespondenties gezocht moet worden) de vorm hebben van sinusoiden. De eerste stereovisie methode die we presenteren parameteriseert de epipolaire curve in termen van een hoek. De gekozen parameterisatie leent zich voor het verkrijgen van diepteschattingen uit een tweetal beelden. Een nadeel bij gebruik van slechts twee beelden is dat het resulterende geschatte dieptebeeld veel fouten bevat. Daarnaast is het onmogelijk om betrouwbare diepteschattingen in de verplaatsingsrichting tussen de twee camera poses te verkrijgen. Om deze beperkingen te het hoofd te bieden stellen we een tweede stereovisie methode voor. We leiden een parameterisatie van epipolaire curven af in termen van een grootheid welke direct is gerelateerd aan diepte. Deze parameterisatie maakt het mogelijk om op eenvoudige wijze diepteschattingen verkregen uit meerde beeldparen met elkaar te verenigen tot een betrouwbaardere diepteschatting.

De reeds besproken methoden veronderstellen dat de gebruikte beelden verkregen zijn op bekende camera poses. We presenteren een twee-staps methode om automatisch relatieve camera poses uit een beeldpaar te schatten. De voorkennis die we daarbij veronderstellen is dat de camera zich parallel aan een (grond)vlak beweegt. In de eerste stap worden de relatieve oriëntatie en richting van verplaatsing geschat uit puntcorrespondenties (waarbij we gebruik maken van een virtuele cilindrische camera). In de tweede stap wordt de lengte van verplaatsing geschat op basis van beelden van het grondvlak (waarbij we gebruik maken van een virtuele gangbare die loodrecht op het grondvlak kijkt). De eerste stap resulteert in praktijk in robuuste schattingen van de relatieve oriëntatie en richting van verplaatsing tussen twee camera poses. De tweede stap, het schatten van de lengte van de verplaatsing, blijkt in simulatie robuust onder aanwezigheid van speculaire reflecties en objecten welke uit het grondvlak steken. We passen de voorgestelde methode toe om een afgelegd traject te reconstrueren uit een sequentie van beelden welke tijdens navigatie zijn vergaard. De relatieve pose schattingen uit (in tijd) opeenvolgende beelden kunnen op eenvoudige wijze worden gecombineerd tot een reconstructie van het afgelegde traject. Op deze manier wordt een vorm van visuele odometrie verkregen. Evenals wiel odometrie, is deze vorm van visuele odometrie onderhevig aan accumulatie van (schattings)fouten. We laten zien dat het effect van de propagatie van schattingsfouten tegengegaan kan worden door tevens beelden te gebruiken die naburig zijn in ruimte, maar niet noodzakelijk in tijd.

De waarde van de in dit proefschrift gepresenteerde methoden en algoritmen wordt getoetst aan de praktijk, en wordt onderschreven door experimenten die zijn uitgevoerd op in praktijk vergaarde camera beelden.

Dankwoord[†]

Allereerst gaat mijn dank uit naar mijn promotor Professor Groen. Zijn eeuwig optimisme en heldere op- en aanmerkingen zijn de inhoud en leesbaarheid van dit boekje zonder enige twijfel ten goede gekomen.

Ben, mijn dagelijks begeleider, ben ik grote dank verschuldigd. Niet alleen voor de inspirerende discussies, maar vooral om de manier waarop hij het onderzoek steeds weer in rechte banen wist te leiden. Aan de trips naar Stockholm, Seoel en Tokio koester ik leuke herinneringen.

Ik dank al mijn fijne collega's, en zonder dat ik een van hen te kort wil doen, zijn er een aantal die ik in het bijzonder wil noemen.

Met Nikos, Edwin en Bas heb ik prettig samengewerkt binnen het RWC project. Nikos, dank voor de verschafte inzichten in probabilistische methoden. Bas, zonder jou had ik (nog) veel meer kostbare uurtjes moeten hacken aan robot en camera software. Edwin, tof dat de robot altijd up-and-running was!

Speciale dank gaat tevens uit naar Emiel Corten. In de beginfase van mijn promotie heb ik in Emiel een fijne collega gevonden. Zijn enthousiasme en programmeerkunsten hebben voor een ware kickstart gezorgd en de programmatuur van zijn hand is voor mij nog altijd een bron van inspiratie. Emiel, ik zal je niet vergeten.

Rien en Kees en ik; een onafscheidelijk trio. Althans, totdat Kees naar het noorden vertrok, en Rien even daarna naar het zuiden. We hebben in de afgelopen jaren heel wat blikjes gespaard, iets meer pintjes gedronken, en veel meer whiteboard pennen leeggekalkt. Thanx voor alle ideeën, suggesties en oplossingen!

Als ik oplossingen noem, mag ik ook vooral Joris niet vergeten. Altijd een luisterend oor en altijd boordevol ideeën, zowel in praktische zin (als locale Matlab/LaTeX guru) als in wetenschappelijke zin. Joris, Josep, Mathijs en Stephan, thanks for discovering and pointing out (surprisingly many) minor glitches in the manuscript.

Mijn naaste vrienden Koen en Jolanda, Dagmar en Mark dank ik voor het feit dat het als ik met jullie was, het eens niet over mijn promotie ging. Daan, na een avondje squashen met jou was ik steeds weer helemaal opgeladen. Eerst alle tegenslagen en bijbehorende

[†] Acknowledgements in Dutch.

frusties (die nu eenmaal met onderzoek gepaard gaan) er flink uit meppen, en dan een ijskoude icetea met een verhitte peptalk!

Ik bedank mijn ouders voor het feit dat zij me altijd hebben gesteund en gestimuleerd om door te leren (en door te bijten!). Mijn zusje en Sicko dank ik vooral voor die ene onvergetelijke dag. Mijn schoonmoeder Wil dank ik voor al haar begrip ("wat doe je dan de hele dag?") en de schik die we nog altijd hebben. Ook Sador mag ik niet vergeten: "effe relaxen baas!". Hoewel het inmiddels een dagelijks ritueel is geworden geniet ik nog steeds volop als we samen uit gaan waaien.

Het leven van een AiO is niet alleen rozegeur en maneschijn. Zonder twijfel is mijn grootste verlies het plotselinge overlijden van mijn schoonvader. Jan, ik mis je. Jouw korte "sterkte", de dag voordat je ons verliet, heeft me extra moed gegeven om de zware laatste loodjes van mijn promotie door te komen.

Helma, woorden schieten tekort. . . Ik heb veel respect voor de manier waarop je hebt om weten te gaan met mijn "promotie-gaat-voor-alles" houding. Ik beloof je lieverd, vanaf morgen ben jij weer #1!