Univerza v Ljubljani

Fakulteta za računalništvo in informatiko

Peter Peer

Gradnja globinskih panoramskih slik v realnem času z uporabo standardnih kamer

Doktorska disertacija

Mentor: prof. dr. Franc Solina

Ljubljana, 2003

University of Ljubljana

Faculty of Computer and Information Science

Peter Peer

Real Time Panoramic Depth Imaging Using Standard Cameras

Doctoral Dissertation

Supervisor: Prof. Dr. Franc Solina

Ljubljana, 2003

Real Time Panoramic Depth Imaging Using Standard Cameras

Peter Peer

Supervisor: prof. dr. Franc Solina

Abstract

A computer vision is a special kind of scientific challenge as we are all users of our own vision systems. Our vision is definitely a source of the major part of information we acquire and process each second. A stereo vision is perhaps even greater challenge, since our own vision system is a stereo one and it performs a complex task, which supplies us with 3D information on our surroundings in a very effective way.

Making machines see is a difficult problem. On one side we have psychological aspects of human visual perception, which try to explain how the visual information is processed in the human brain. On the other side we have technical solutions, which try to imitate human vision. Normally, it all starts with capturing digital images that store the basic information about the scene in a similar way that humans see. But this information represents only the beginning of a difficult process. By itself it does not reveal the information about the objects on the scene, their color, distances etc. to the machine. For humans, visual recognition is an easy task, but the human brain processing methods are still a mistery to us.

One part of the human visual perception is estimating the distances to the objects on the scene. This information is also needed by robots if we want them to be completely autonomous.

In this dissertation we present a stereo panoramic depth imaging system.

The basic system is mosaic-based, which means that we use a single standard rotating camera and assemble the captured images in a multiperspective panoramic image. Due to a setoff of the camera's optical center from the rotational center of the system we are able to capture the motion parallax effect, which enables the stereo reconstruction. The camera is rotating on a circular path with the step defined by an angle equivalent to one-pixel column of the captured image. To find the corresponding points on a stereo pair of panoramic images the epipolar geometry needs to be determined. It can be shown that the epipolar geometry is very simple if we perform the reconstruction based on a symmetric pair of stereo panoramic images. We get a symmetric pair of stereo panoramic images when we take symmetric columns on the left and on the right side from the captured image center column.

This system however cannot generate panoramic stereo pair in real time. That is why we have suggested a real time extension of the system, based on simultaneously using many standard cameras. We have not physically built the real time sensor, but we have performed simulations to establish the quality of results.

Both systems have been comprehensively analysed and compared. The analyses revealed a number of interesting properties of the systems. According to the basic system accuracy we definitely can use the system for autonomous robot localization and navigation tasks. The assumptions made in the real time extension of the basic system have been proved to be correct, but the accuracy of the new sensor generally deteriorates in comparison to the basic sensor.

Generally speaking, the dissertation can serve as a guide for panoramic depth imaging sensor design and related issues.

Key words

computer vision, stereo vision, reconstruction, depth image, multiperspective panoramic image, mosaicing, motion parallax effect, standard camera, real time, depth sensor

Contents

List of Figures

List of Tables

To my family

Chapter 1

Introduction

1.1 Description of the narrow scientific area

In the last 30 years one of the most interesting areas of research is building machines that would complement human life with the help of artificial intelligence. This area is full of different challenges and one among them is to imitate human vision. Analogically this discipline is called computer vision. The basic idea is to discover properties of 3D world by using only 2D information from an image. A lot of effort was put into this area of research, which eventually led to progress in areas such as object recognition, picture understanding and 3D reconstruction.

3D reconstruction is an important area, since it enables tasks like modeling, visualization, CAD (Computer Aided Design) model construction, localization, navigation etc. Generally speaking 3D shape of objects and environments can be captured in three different ways: using CMM (Coordinate Measuring Machine) machine, Time of Flight method or with the help of optical devices. The latter approach is most widely used, partly because of favorable price and safety conditions. On the other hand such a system is the only real computer vision system for 3D reconstruction, since it is based merely on input images. With the help of optical scanners (range finders) we gather 3D data about the object surface by processing 2D images captured with standard cameras. Optical scanners can be divided into two main groups. Active scanners project light on the object, which assures effective and reliable 3D information. In many cases active scanners use structured light for reconstruction purpose, i.e. a light pattern is projected onto the object. A disadvantage of such a scanner is that the images should be taken under strict laboratory conditions, like scanning in complete darkness. On the other side we have passive scanners that estimate the distance to the object only on present textural information on images, which are captured completely without contact (physical contact, contact of a laser beam or structured light). Traditional approaches for acquiring depth from such images are based on stereo methods. Under the term stereo reconstruction we understand the generation of depth images from two or more captured images. In this case the reconstruction suffers if the reconstructed object is not well textured. The result of the reconstruction is a depth image. Each depth image stores the estimates of distances to the object from one view point.

Currently, we are living in an era of vision research when some shape-from-X problems, for example stereo, have been almost completely solved, and furthermore are being used in industry. Other shape-from-X problems in their original formulations, like shape from motion, have proved to be very difficult, therefore some special cases are being tackled. The remaining shape-from-X problems, like shape from shading and shape from texture, have become less interesting, and, what is even more important, less applicable [50].

We wish that input images would have the property that the same points and lines are visible in all images of the scene, which facilitate stereo reconstruction. This is the property of panoramic cameras. Standard cameras have a limited field of

view, which is usually smaller than the human field of view. Because of that people have always tried to generate images with a wider field of view, up to full 360 degree panoramas.

As presented in the next chapter, one way to build panoramic images is by taking one column out of a captured image and mosaicing the columns. Such panoramic images are called multiperspective panoramic images. The crucial property of two or more multiperspective panoramic images is that they capture the information about the motion parallax effect, since the columns forming the panoramic images are captured from different perspectives.

The main problem we would like to solve in this dissertation is to analyze and determine the properties and the efficiency of the panoramic depth imaging system based on multiperspective panoramic images and see if these results can be used for robot localization and navigation. Only standard equipment is applied in the system construction process. A real time extension of the basic system is simulated to determine the efficiency of the new system in comparison to the basic one.

When we talk about the real time in the dissertation, we do not mean it so much from the processing power point of view (to increase the speed or reduce the time), but more from the accuracy point of view. Namely, nowadays there are many practical solutions how to increase processing power, but before we invest in the real system, we have to know whether the accuracy of the system is satisfactory. Nevertheless, the processing power is also briefly addressed in the dissertation.

1.2 Description of the problem

For effective depth reconstruction we need high resolution images. As described in the next chapter, only mosaic-based procedures give high resolution results. Thus these procedures represent a good starting point for the development of our system.

First of all, we are interested in where are the efficiency borders of stereo panoramic depth imaging system, which is based on multiperspective panoramic images. The basic system consists of only one standard camera, which is offset from the system's rotational center. It is rotated around the rotational center in angular steps corresponding to one vertical pixel-column of the captured standard image. In this way the best possible accuracy of the depth reconstruction process is achieved and consequently the results of this system serve as a ground truth for subsequent comparisons.

Therefore the focus of the first part of the dissertation (Chapter 2) is on the exposed research issues. In it we also prove that we can effectively constrain the search space on the epipolar line, that the confidence in the estimated depth is variable and that the system can be used for depth reconstruction of small rooms (having in mind an application to autonomous mobile robot navigation). The relationship between different system parameters is also presented.

The disadvantage of the mosaicing procedures lies in the time needed to capture many images. Therefore we suggest a real time extension of the system based on simultaneously using many standard cameras. Only real time execution ensures the possibility to reconstruct dynamic scenes, which is in many cases of great importance for autonomous mobile robot navigation.

The second part of the dissertation (Chapter 3) explains this solution in detail. It reveals a new panoramic depth sensor, its properties, analyze the number of needed standard cameras in order to ensure a good compromise between the speed and the accuracy of the new sensor, because the panoramic images are built from wider stripes and not from only one column of the captured image, and new results are compared to the results of the basic sensor, described in the first part of the dissertation.

As mentioned, suggested real time panoramic depth sensor can only be build if we use an adequate number of cameras simultaneously. We geometrically prove that such a sensor can be build out of standard cameras, which are available on the market. We have not physically built it, but we have performed simulations to establish the quality of results.

The in-depth analysis of such mosaicing approach should reveal whether the system could be used for real time panoramic depth imaging and consequently for autonomous mobile robot navigation.

Since our final goal is to determine the usability of our system for mobile robot navigation, we perform all the tests on real world images, so that the results reflect the applicability of implemented algorithms in the real world.

1.3 Structure of the dissertation

Basic introduction to the field of computer vision, related to the title of the dissertation, and the problem statement are given in this chapter. In the next chapter (part one of the dissertation) the basic system based on a camera mounted on a rotational arm so that the optical center of the camera is offset from the vertical axis of rotation is introduced and evaluated. The descriptions of different panoramic cameras and related work are also part of this chapter. In Chapter 3 (second part of the dissertation), we suggest a real time extension of the basic system, reveal its properties and evaluate its effectiveness, also in comparison to the basic system. The summary of the dissertation is given in the last chapter, along with the conclusions, the contributions to science and the ideas for future work. We end the dissertation with the extended summary in Slovenian language, which is given in the appendix.

Chapter 2

Basic System

2.1 Introduction

2.1.1 Motivation

Standard cameras have a limited field of view, which is usually smaller than the human field of view. Because of that people have always tried to generate images with a wider field of view, up to full 360 degree panorama [16].

Under the term stereo reconstruction we understand the generation of depth images from two or more captured images. A depth image is an image that stores distances to points on the scene. The stereo reconstruction procedure is based on relations between points and lines on the scene and images of the scene. If we want to get a linear solution of the reconstruction procedure then the images can interact with the procedure in pairs, triplets or quadruplets, and relations are named accordingly to the number of images as epipolar constraint, trifocal constraint or quadrifocal constraint [22]. We want the images to have the property that the same points and lines are visible in all images of the scene, which facilitate stereo reconstruction. This is the property of panoramic cameras and it presents our fundamental motivation. The stereo reconstruction in this dissertation is done from two symmetric multiperspective panoramic images.

In this work we address only the issue how to enlarge the horizontal field of view of images. The vertical field of view of panoramic images can be enlarged by using wide angle camera lenses [44], by using mirrors [25, 32] or by moving the camera also in the vertical direction and not only in the horizontal direction [16].

If we tried to build two panoramic images simultaneously by using two standard cameras which are mounted on two rotational robotic arms, we would have problems with non-static scenes. Clearly, one camera would capture the motion of the other camera. So we have decided to use one camera only. In the first part of our work we develop a mosaic-based panoramic depth imaging system using only one standard camera and analyze its performance to see if it can be used for robot localization and navigation in a room.

2.1.2 Basics about the system

In Fig. 2.1 the hardware part of our system can be seen: a color camera is mounted on a rotational robotic arm so that the optical center of the camera is offset from the vertical axis of rotation. The camera is looking outward from the system's rotational center. Panoramic images are generated by repeatedly shifting the rotational arm by an angle which corresponds to a single pixel column of the captured image. By assembling the center columns of these images, we get a mosaic panoramic image. One of the drawbacks of mosaic-based panoramic imaging is that dynamic scenes are not well captured.

It can be shown that the epipolar geometry is very simple if we perform the reconstruction based on a symmetric pair of stereo panoramic images. We get a

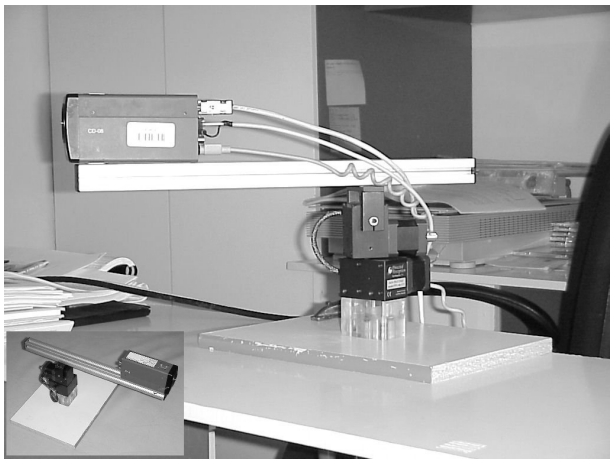


Figure 2.1: Hardware part of our system.

symmetric pair of stereo panoramic images when we take symmetric columns on the left and on the right hand side from the captured image center column. These columns are assembled in a mosaic stereo pair. The column from the left hand side of the captured image is mosaiced in the right eye panoramic image and the column from the right hand side of the captured image is mosaiced in the left eye panoramic image.

2.1.3 Structure of the chapter

In the next section we compare different panoramic cameras with emphasis on mosaicing. In Sec. 2.3 we give an overview of related work and briefly present the contribution of our work towards the discussed subject. Sec. 2.4 describes the geometry of our system, Sec. 2.5 is devoted to the epipolar geometry and Sec. 2.6 describes the procedure of stereo reconstruction. The focus of this chapter is on the analysis of system capabilities, given in Sec. 2.7. In Sec. 2.8 we present experimental results. In the very end of this chapter we summarize the main conclusions of the first part of the dissertation.

2.2 Panoramic cameras

Every panoramic camera belongs to one of three main groups of panoramic cameras: catadioptric cameras, dioptric cameras and cameras with moving parts. The basic property of a catadioptric camera is that it consists of a mirror (or mirrors [18]) and a camera. The camera captures the image which is reflected from the mirror. A dioptric camera is using a special type of lens, e.g. fish-eye lens, which increases the size of the camera's field of view. A panoramic image can also be generated by

moving the camera along some path and mosaicing together the images captured in different locations on the path.

Type of panoramic camera	Number of images	Resolution of panoramic images	Real time	References
catadioptric camera	1	low	yes	[15, 18, 25, 28, 29, 33, 52]
dioptric camera	1	low	yes	[3, 7]
moving parts	a lot	high	no	[1, 8, 9, 10, 12, 13, 14, 16] [17, 19, 20, 21, 23, 25, 26] [27, 32, 33, 35, 36, 39, 43] [44]

Table 2.1: Comparison of different types of panoramic cameras with respect to the number of standard images needed to build a panoramic image, the resolution of panoramic images and the capability of building a panoramic image in real time.

The comparison of different types of panoramic cameras is shown in Tab. 2.1.

All types of panoramic cameras enable 3D reconstruction. The camera has a single viewpoint or a projection center if all light rays forming the image intersect in a single point. Cameras with this property are also called central cameras. Rays forming a non-central image do not pass through a single point, but rather intersect a line [10], a conic [25, 39, 40, 49], do not intersect at all [46] or are bound by other constraints suiting the practical or the theoretical demands [13, 17].

Mosaic-based procedures can be marked as non-central (we do not deal with a single center of projection), they do not execute in real time, but they give high resolution results. High resolution images enable effective depth reconstruction, since by increasing the resolution the number of possible depth estimates is also increasing. Thus mosaicing is not appropriate for capturing dynamic scenes and consequently not for reconstruction of dynamic scenes. The systems described in [1, 16] are exceptions because the light rays forming the mosaic panoramic image intersect in the rotational center of the system. These two systems are central systems. The system presented in [30, 41, 42] could also be treated as mosaic-based procedure, though its concept for generating panoramic depth images is very different from our concept. Because the system is more related to the topic of the second part of the dissertation (Chapter 3), it is presented in Sec. 3.1.1.

Dioptric panoramic cameras with wide angle lenses can be marked as non-central [29], they build a panoramic image in real time and they give low resolution results. Cameras with wide angle lenses are appropriate for fast capturing of panoramic images and processing of captured images, e.g. for detection of obstacles or for localization of a mobile robot, but are less appropriate for reconstruction. Please note that we are talking about panoramic cameras here. Generally speaking, dioptric

cameras can be central.

Only some of the catadioptric cameras have a single viewpoint. Cameras with a mirror (or mirrors) work in real time and they give low resolution results. Only two mirror shapes, namely hyperbolic and parabolic mirrors, can be used to construct a central catadioptric panoramic camera [29, 52]. Such panoramic cameras are appropriate for low resolution reconstruction of dynamic scenes and for motion estimation. It is also true that only for panoramic systems with hyperbolic and parabolic mirrors the epipolar geometry can be simply generalized [29, 52].

Since dioptric and catadioptric cameras give low resolution results, they are more appropriate for use with view-based systems [59] and less for use with reconstruction systems.

Of course, combinations of different cameras exist: e.g. a combination of the mosaicing camera and the catadioptric camera [25, 32] or a combination of the mosaicing camera and the wide angle camera [44]. Their main purpose is to enlarge the camera's vertical field of view.

2.3 Related work

We can generate panoramic images either with the help of special panoramic cameras or with the help of a standard camera and with mosaicing standard images into panoramic images. If we want to generate mosaic 360 degree panoramic images, we have to move the camera on a closed path, which is in most cases a circle.

One of the best known commercial packages for creating mosaic panoramic images is QTVR (QuickTime Virtual Reality). It works on the principle of sewing together a number of standard images captured while rotating the camera [8]. Peleg et al. [27] introduced the method for creation of mosaiced panoramic images from standard images captured with a handheld video camera. A similar method was suggested by Szeliski and Shum [12], which also does not strictly constraint the camera path but assumes that a great motion parallax effect is not present. All methods mentioned so far are used only for visualization purposes since the authors did not try to reconstruct the scene.

The crossed-slits (X-slits) projection [53, 56, 61] uses a similar mosaicing technique with one important difference: the mosaiced strips are sampled from varying positions in the captured images. This makes the generation of virtual walkthroughs possible, i.e. we are again dealing with the visualization with the help of image-based rendering or new view synthesis.

Ishiguro et al. [1] suggested a method which enables scene reconstruction. They used a standard camera rotating on a circular path. The scene is reconstructed by means of mosaicing panoramic images together from the central column of the captured images and moving the system to another location where the task of mosaicing is repeated. The two created panoramic images are then used as the input

to a stereo reconstruction procedure. The depth of an object was first estimated using projections in two images captured in different locations of the camera on the camera path. But since their primary goal was to create a global map of the room, they preferred to move the system attached to the robot about the room. Clearly, by moving the robot to another location and producing the second panoramic image of a stereo pair in this location rather than producing a stereo pair in a single location, they enlarged the disparity of the system. But this decision also has a few drawbacks: we cannot estimate the depth for all points on the scene, the time of capturing a stereo pair is longer and we have to search for the corresponding points on the sinusoidal epipolar curves. The depth was then estimated from two panoramic images taken at two different locations of the robot in the room.

Peleg and Ben-Ezra [19, 26] introduced a method for creation of stereo panoramic images without actually computing the 3D structure — the depth effect is created in the viewer's brain.

In [20], Shum and Szeliski described two methods used for creation of panoramic depth images, which use standard procedures for stereo reconstruction. Both methods are based on moving the camera on a circular path. Panoramic images are built by taking one column out of a captured image and mosaicing the columns. The authors call such panoramic images *multiperspective panoramic images*. The crucial property of two or more multiperspective panoramic images is that they capture the information about the motion parallax effect, since the columns forming the panoramic images are captured from different perspectives. The authors use such panoramic images as the input in a stereo reconstruction procedure. In [21], Shum et al. proposed a non-central camera called an omnivergent sensor in order to reconstruct scenes with minimal reconstruction error. This sensor is equivalent to the sensor presented in this chapter.

However, multiperspective panoramic images are not something new to the vision community [20]: they are a special case of *multiperspective panoramic images for cel animation* [13], a special case of *crossed-slits (X-slits) projection* [53, 56, 61], they are very similar to images generated by a procedure called *multiple-center-of-projection* [17], by the *manifold projection* procedure [27] and by the *circular projection* procedure [19, 26]. The principle of constructing multiperspective panoramic images is also very similar to the *linear pushbroom camera* principle for creating panoramic images [10].

The papers closest to our work [1, 20, 21] seem to lack two things: a comprehensive analysis of 1) the system's capabilities and 2) the corresponding points search using the epipolar constraint. Therefore, the focus of this chapter is on these two issues. While in [1] the authors searched for corresponding points by tracking the feature from the column building the first panorama to the column building the second panorama, the authors in [20] used an upgraded *plane sweep stereo* procedure. A key idea behind the approach in [21] is that it enables optimizing the input to traditional computer vision algorithms for searching the correspondences in order to

produce superior results.

Further details about the related work are revealed in in the following sections, where we discuss specifics of our system.

2.4 System geometry

Let us begin this section with description of how the stereo panoramic pair is generated. From the captured images on the camera's circular path we always take only two columns, which are equally distant from the middle column. We assume that the middle column that we are referring to in this work, is the middle column of the captured image, if not mentioned otherwise. The column on the right hand side of the captured image is then mosaiced in the left eye panoramic image and the column on the left hand side of the captured image is mosaiced in the right eye panoramic image. So, we are building each panoramic image from just a single pixel column of the captured image. Thus, we get a symmetric pair of stereo panoramic images, which yields a reconstruction with optimal characteristics (simple epipolar geometry and minimal reconstruction error) [21].

The geometry of our system for creating multiperspective panoramic images is shown in Fig. 2.2. The panoramic images are then used as the input to create panoramic depth images. Point C denotes the system's rotational center around which the camera is rotated. The offset of the camera's optical center from the rotational center C is denoted as r, describing the radius of the circular path of the camera. The camera is looking outward from the rotational center. The optical center of the camera is marked with O. The column of pixels that is sewn in the panoramic image contains the projection of point P on the scene. The distance from point P to point C is the depth l, while the distance from point P to point O is denoted by d. Further, θ is the angle between the line defined by points C and O and the line defined by points C and P. In the panoramic image the horizontal axis represents the path of the camera. The axis is spanned by μ and defined by point C, a starting point O_0, where we start capturing the panoramic image, and the current point O. φ denotes the angle between the line defined by point O and the middle column of pixels of the image captured by the physical camera looking outward from the rotational center (the latter column contains the projection of the point Q), and the line defined by point O and the column of pixels that will be mosaiced into the panoramic image (the latter column contains the projection of the point P). Angle φ can be thought of as a reduction of the camera's horizontal view angle α.

The geometry of capturing multiperspective panoramic images can be described with a pair of parameters (r, φ). By increasing (decreasing) each of them, we increase (decrease) the baseline ($2r_0$ [39], $r_0 = r \cdot \sin\varphi$ (Fig. 2.2)) of our stereo system.

Wei et al. [43] proposed an approach to solve the parameter (r, φ) determination problem for a symmetric stereo panoramic camera. The image acquisition parame-

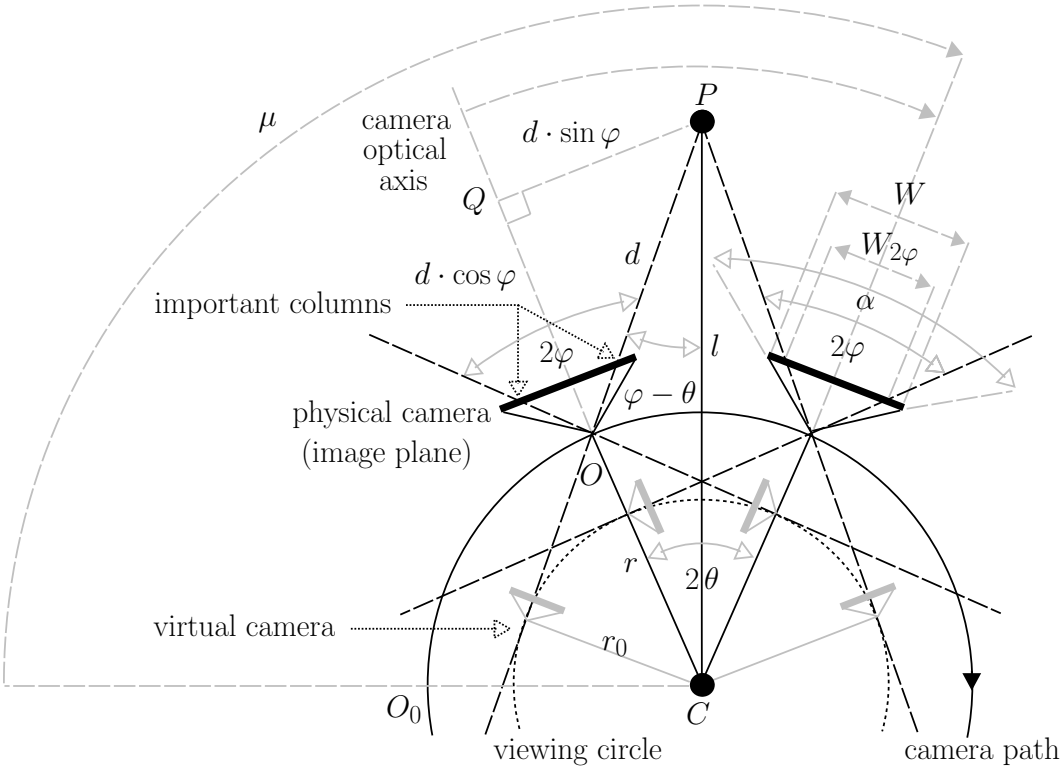


Figure 2.2: Geometry of our system for constructing multiperspective panoramic images. Note that a ground-plan is presented. The optical axis of the camera is kept horizontal.

ters (r, φ) are calculated based on (subjectively) given parameters: the nearest and the furthest distances of the region of interest, the height of the region of interest and the width of the angular disparity interval. They conclude that neither the parameter r nor φ can satisfactorily match application requirements on their own and report that a general study of relations among parameters is in progress as they have discovered certain exceptions in experiments that require further researches.

The system in Fig. 2.2 is obviously a non-central since the light rays forming the panoramic image do not intersect in one point called the viewpoint, but instead are tangent ($\varphi \neq 0$) to a cylinder with radius r_0, called the viewing cylinder (Fig. 2.3). Thus, we are dealing with panoramic images formed by a projection from a number of viewpoints. This means that a captured point on the scene is seen in the panoramic image from one viewpoint only. This is why the panoramic images captured in this way are called multiperspective panoramic images.

For stereo reconstruction we need two images. If we look at only one circle on the viewing cylinder (Fig. 2.2) then we can conclude that our system is equivalent

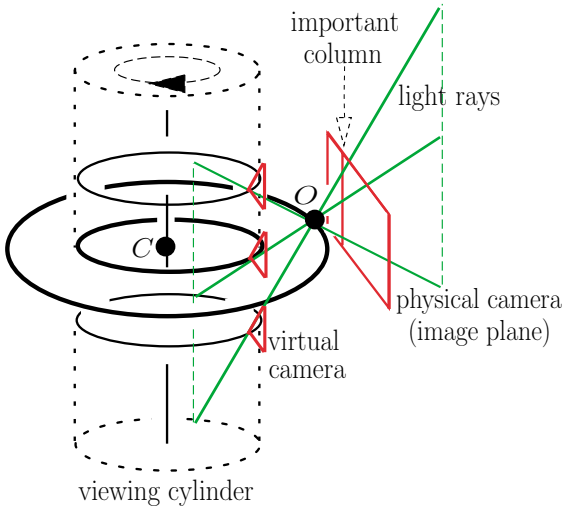


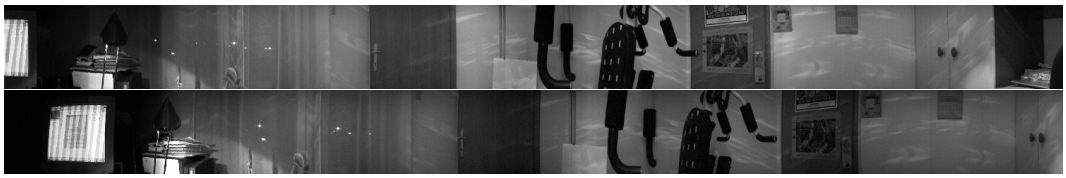
Figure 2.3: All the light rays forming the panoramic image are tangent to the viewing cylinder.

to a system with two cameras. In our case, two virtual cameras are rotating on a circular path, i.e. a viewing circle, with radius r_0. The optical axis of a virtual camera is always tangent to the viewing circle. The panoramic image is generated from only one pixel from the middle column of each image captured by a virtual camera. This pixel is determined by the light ray which describes the projection of a scene point on the physical camera image plane. If we observe a point on the scene P, we see that both virtual cameras, which see this point, form a traditional stereo system of converging cameras.

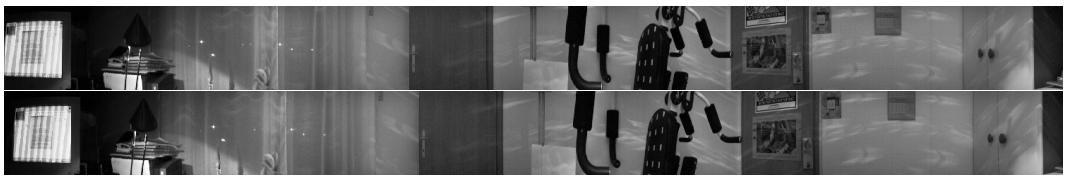
Obviously, a symmetric pair of panoramic images used in the stereo reconstruction process could be captured also with a bunch of cameras rotating on a circular path with radius r_0, where the optical axis of each camera is tangent to the circular path (Fig. 2.3).

Two images differing in the angle of rotation of the physical camera setup (for example, two image planes marked in Fig. 2.2) are used to simulate a bunch of virtual cameras on the viewing cylinder. Each column of the panoramic image is obtained from a different position of the physical camera on a circular path. In Fig. 2.4 we present two symmetric pairs of panoramic images.

To automatically register captured images directly from the knowledge of the camera's viewing direction, the camera lens' horizontal view angle α and vertical view angle β are required. If we know this information, we can calculate the resolution of one angular degree, i.e. we can calculate how many columns and rows are within an angle of one degree. The horizontal view angle is especially important



$2\varphi = 29.9625°$



$2\varphi = 3.6125°$

Figure 2.4: Two symmetric pairs of panoramic images generated using different values of the angle φ. In Sec. 2.7.1 we explain where these values for the angle φ come from. Each symmetric pair of panoramic images comprises the motion parallax effect. This fact enables the stereo reconstruction.

in our case, since we move the rotational arm only around it's vertical axis. To calculate these two parameters, we use an algorithm described in [16]. It is designed to work with cameras whose zoom settings and other internal camera parameters are unknown. The algorithm is based on the mechanical accuracy of the rotational arm. The basic step of our rotational arm corresponds to an angle of $0.0514285\overline{7}\,°$. In general, this means that if we tried to turn the rotational arm for 360 degrees, we would perform 7000 steps. Unfortunately, the rotational arm that we use cannot turn for 360 degrees around it's vertical axis. The basic idea of the algorithm is to calculate the translation dx (in pixels) between two images, while the camera is rotated for a known angle $d\gamma$ in the horizontal direction. Since we know the exact angle by which we move the camera, we can calculate the horizontal view angle of the camera:

$$\alpha = \frac{W}{dx} \cdot d\gamma, \tag{2.1}$$

where W is the width of the captured image in pixels.

The major drawback of this method is that it relies on the accuracy of the rotational arm. Because of that we rechecked the values of the view angles by calibrating the camera using a static camera and a checkboard pattern [11, 31, 54]. The input into the calibration procedure is a set of images with varying position of the pattern in each image. The results obtained were very similar, though the second method should be more reliable as it reveals more information about the camera model and also uses sub-pixel accuracy procedure. The latter calibration estimates the focal length, the principal point, the skew coefficient and distortions,

to name just the most important parameters for us. It also reveals the errors of all estimated parameters. If we assume that the principal point is in the middle of the captured image, we can calculate the horizontal view angle of the camera from the estimated parameters:

$$\alpha = 2 \arctan \frac{W/2}{f}, \tag{2.2}$$

where f is the estimated focal length.

Distortion parameters are also important, because we also investigate the influence of distortion on the system's results.

In any case, now that we know the value of α, we can calculate the resolution of one angular degree x_0:

$$x_0 = \frac{W}{\alpha}.$$

This equation enables us to calculate the width of the stripe W_s that will be mosaiced in the panoramic image when the rotational arm moves for an angle θ_0:

$$W_s = x_0 \cdot \theta_0. \tag{2.3}$$

From the above equation we can also calculate the angle of the rotational arm for which we have to move the rotational arm if the stripe is only one pixel column wide.

We used three different cameras in the experiments:

- a camera with the horizontal view angle $\alpha = 34°$ and the vertical view angle $\beta = 25°$,
- a camera with the horizontal view angle $\alpha = 39.72°$ and the vertical view angle $\beta = 30.54°$,
- a camera with the horizontal view angle $\alpha = 16.53°$ and the vertical view angle $\beta = 12.55°$.

In the process of the panoramic image construction we did not vary these two parameters. From here on, the first camera is used in the calculations and the experiments, if not stated differently.

2.5 Epipolar geometry

Searching for the corresponding points in two images is a difficult problem. Generally speaking, the corresponding point can be anywhere in the second image. That is why we would like to constrain the search space as much as possible. Using the epipolar constraint we reduce the search space from 2D to 1D, i.e. to an epipolar

line [4]. In Sec. 2.7.3 we prove that in our system we can effectively reduce the search space even on the epipolar line.

In this section we will only illustrate the procedure of the proof that the epipolar lines of the symmetric pair of panoramic images are image rows. This statement is true for our system geometry. For proof see [20, 23, 35, 51].

The proof in [23] is based on radius r_0 of the viewing cylinder (Figs. 2.2 and 2.3). We can express r_0 in the terms of known parameters r and φ as:

$$r_0 = r \cdot \sin \varphi \ .$$

We carry out the proof in three steps: *first*, we have to execute the projection equation for the line camera, *then* we have to write the projection equation for a multiperspective panoramic image and, in the *final* step, we prove the property of the epipolar lines for the case of a symmetric pair of panoramic images. In the first step, we are interested in how the point on the scene is projected to the camera's image plane [4], which is of dimension $n \times 1$ pixels in our case, since we are dealing with a line camera. In the second step, we have to write the relation between different notations of a point on the scene and the projection of this point on the panoramic image: notation of the scene point in Euclidean coordinates of the world coordinate system and in cylindric coordinates of the world coordinate system, notation of the projected point in angular coordinates of the (2D) panoramic image coordinate system and in pixel coordinates of the (2D) panoramic image coordinate system. When we know the relations between the above-mentioned coordinate systems, we can write the equation for projection of scene points on the cylindric image plane of the panorama. Based on the angular coordinates of the panoramic image coordinate system property, we can in the third step show that the epipolar lines of the symmetric pair of panoramic images are actually rows of panoramic images. The basic idea for the last step of the proof is as follows: If we are given an image point in one panoramic image, we can express the optical ray defined by a given point and the optical center of the camera in 3D world coordinate system. If we project this optical ray described in world coordinate system on the second panoramic image, we get an epipolar line corresponding to the given image point in the first panoramic image. After introducing proper relations valid for the symmetric case into the obtained equation, our hypothesis is confirmed.

The same result can be found in [20], where the authors proved the property of symmetric pair of panoramic images by directly investigating the presence of the vertical motion parallax effect in the panoramic images captured from the same rotational center. The generalization to the non-symmetric case for the camera looking inward and outward can be found in [51]. Even a more general case, in some respect, where the panoramic images can be captured from different rotational centers, is discussed in [35].

It was shown that the notion of the epipolar geometry, well known for both central perspective cameras [4, 22, 34] and central catadioptric cameras [28, 29, 52],

can be generalized to some non-central cameras [37, 40, 46, 49]. The epipolar surfaces extend from planes to double-ruled quadrics: planes, rotational hyperboloids and hyperbolic paraboloids.

2.6 Stereo reconstruction

Let us go back to Fig. 2.2. Using trigonometric relations evident from the sketch, we can write the equation for the depth estimation l of a point P on the scene. By the basic law of sines for triangles, we have:

$$\frac{r}{\sin(\varphi-\theta)} = \frac{d}{\sin\theta} = \frac{l}{\sin(180^\circ-\varphi)}. \tag{2.4}$$

From this equation we can express the equation for depth estimation l as:

$$l = \frac{r\cdot\sin(180^\circ-\varphi)}{\sin(\varphi-\theta)} = \frac{r\cdot\sin\varphi}{\sin(\varphi-\theta)}. \tag{2.5}$$

Eq. (2.5) implies that we can estimate depth l only if we know three parameters: r, φ and θ. r is given. Angle φ can be calculated on the basis of the camera's horizontal view angle α (Eq. (2.1)) as:

$$2\varphi = \frac{\alpha}{W}\cdot W_{2\varphi}, \tag{2.6}$$

where W is the width of the captured image in pixels and $W_{2\varphi}$ is the width of the captured image between columns forming the symmetric pair of panoramic images, given also in pixels. To calculate the angle θ, we have to find corresponding points on panoramic images. Our system works by moving the camera for the angle corresponding to one pixel column of the captured image. If we denote this angle by θ_0, we can express the angle θ as:

$$\theta = dx\cdot\frac{\theta_0}{2}, \tag{2.7}$$

where dx is the absolute value of difference between the corresponding points image coordinates on the horizontal axis x of the panoramic images.

Note that Eg. (2.5) does not contain the focal length f explicitly, but since the relationships between α and f on one side (Eq. (2.2)) and α and φ on the other side (Eq. (2.6)) exist, φ also depends upon f (the two models for estimating angle φ (Eqs. (2.6) and (2.8)) are discussed in Sec. 2.7.2):

$$\varphi = \arctan\frac{W_{2\varphi}/2}{f}. \tag{2.8}$$

Eq. (2.5) estimates the distance l to the perpendicular projection of the scene point P on the plane defined by the camera's circular (planar) path. The projection

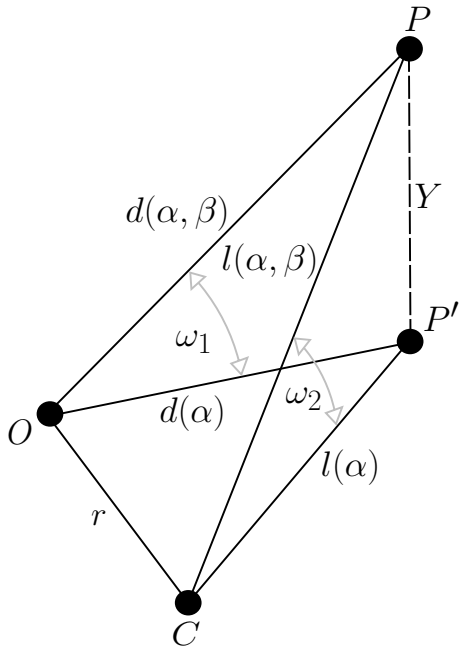


Figure 2.5: Important relations between system parameters for addressing the vertical reconstruction.

of the scene point P is marked with P' in Fig. 2.5. Since this estimation is an approximation of the real l, we have to improve the estimation by addressing the vertical reconstruction, i.e. by incorporating the vertical view angle β into Eq. (2.5).

Let us here adopt the following notation to introduce the influence of β on estimation of l: if a variable l or d depends on α only, we mark that as $l(\alpha)$ and $d(\alpha)$ (until now, these variables were marked simply l and d), but if a variable l or d depends on α and β, we mark that as $l(\alpha, \beta)$ and $d(\alpha, \beta)$. According to Fig. 2.5 the distance to the point P on the scene can be calculated as:

$$l(\alpha, \beta) = \sqrt{l(\alpha)^2 + Y^2} = \sqrt{l(\alpha)^2 + (l(\alpha) \cdot \tan \omega_2)^2}.$$

Because the value of ω_2 is unknown, we have to express it in terms of known parameters. We can do that, while Y can also be written as:

$$Y = d(\alpha) \cdot \tan \omega_1.$$

We can calculate ω_1 similarly as we calculated φ (Eqs. (2.6) and (2.8)):

$$2\omega_1 = \frac{\beta}{H} \cdot H_{2\omega_1} \quad \text{or} \quad \omega_1 = \arctan \frac{H_{2\omega_1}/2}{f},$$

where H is the height of the captured image in pixels and $H_{2\omega_1}$ is the height of the captured image between the image row that contains the projection of the scene point P and the symmetric row on the other side from the middle row, given also in pixels. And $d(\alpha)$ follows from Eq. (2.4):

$$d(\alpha) = \frac{l(\alpha) \cdot \sin\theta}{\sin\varphi}.$$

Now, we can write the equation for $l(\alpha, \beta)$ as:

$$l(\alpha, \beta) = \sqrt{l(\alpha)^2 + \left(\frac{l(\alpha) \cdot \sin\theta}{\sin\varphi} \cdot \tan\omega_1\right)^2}. \tag{2.9}$$

From now on, $l = l(\alpha)$ and when $l(\alpha, \beta)$ is used, this is explicitly stated.

The influence of addressing the vertical reconstruction on the reconstruction accuracy is discussed in Secs. 2.7.6 and 2.8.4.

2.7 Analysis of the system's capabilities

2.7.1 Time complexity of panoramic image creation

The biggest disadvantage of our system is that it cannot produce panoramic images in real time since we create them stepwise by rotating the camera for a very small angle. Because of mechanical vibrations of the system, we also have to ensure to capture an image when the system is completely still. The time that the system needs to create a panoramic image is much too long to allow it work in real time.

In a single circle around the system's vertical axis our system constructs 11 panoramic images: 5 symmetric pairs and a panoramic image from the middle columns of the captured images. It captures and saves 1501 images with resolution of 160×120 pixels, where radius is $r = 30$ cm and the shift angle is $\theta_0 = 0.205714°$. We have choosen the resolution of 160×120 pixels because it represents a good compromise between overall time complexity of the system and its accuracy, as it is shown in the following sections. We cannot capture $360/\theta_0$ images because of the limitation of the rotational arm. Namely, the rotational arm cannot turn for 360 degrees around its vertical axis.

The middle column of the captured image was in our case the 80th column. The distances between the columns building up symmetric pairs of panoramic images were 141, 125, 89, 53 and 17 columns. These numbers include two columns building up each pair. In consequence the values of the angle 2φ (Eq. (2.6)) are 29.9625° (141 columns), 26.5625° (125 columns), 18.9125° (89 columns), 11.2625° (53 columns) and 3.6125° (17 columns), respectively. (Here we used the camera with the horizontal view angle $\alpha = 34°$.)

The acquisition process takes little over 15 minutes on a 350 MHz Intel PII PC. The steps of the acquisition process are as follows:

1. Move the rotational arm to its initial position.
2. Capture and save the image.
3. Contribute image parts to the panoramic images.
4. Move the arm to the new position.
5. Check in the loop if the arm is already in the new position. The communication between the program and the arm is written in the file for debugging purposes. After the program exits the loop, it waits for 300 ms in order to stabilize the arm in the new position.
6. Repeat steps 2 to 5 until the last image is captured.
7. When the last image has been captured, contribute image parts to the panoramic images and save them.

We could achieve faster execution since our code is not optimized. For example, we did not optimize the waiting time (300 ms) after the arm is in the new position. No computations are done in parallel.

2.7.2 Influence of parameters r, φ and θ_0 on the reconstruction accuracy

In order to estimate the depth as precisely as possible, the parameters involved in the calculation also have to be estimated precisely. In this section we reveal the methods used for estimation of parameters r, φ and θ_0.

θ_0 denotes the angle corresponding to one pixel column of the captured image, for which we rotate the camera. It can be calculated from Eq. (2.3):

$$\theta_0 = \frac{\alpha}{W}. \tag{2.10}$$

For $\alpha = 34°$ and W=160 pixels, we get $\theta_0 = 0.2125°$. On the other hand, we know that the accuracy of our rotational arm is $\varepsilon = 0.0514285\overline{5}°$, so the best possible approximate value is $\theta_0 = 0.205714°$. Since each column in the panoramic image in reality describes the latter angle θ_0, we always use in calculations $\theta_0 = n \cdot \varepsilon$, $n \in IN$, which is closest to the result obtained from Eq. (2.10). The experiment in Sec. 2.8.5 confirms that this decision is correct. To discriminate the two values between each other, let us mark them as $\theta_0(\alpha)$ (Eq. (2.10)) and $\theta_0(\varepsilon)$ (the estimation based on the accuracy of our rotational arm). We use them from now on, but where only θ_0 is given, then $\theta_0 = \theta_0(\varepsilon)$.

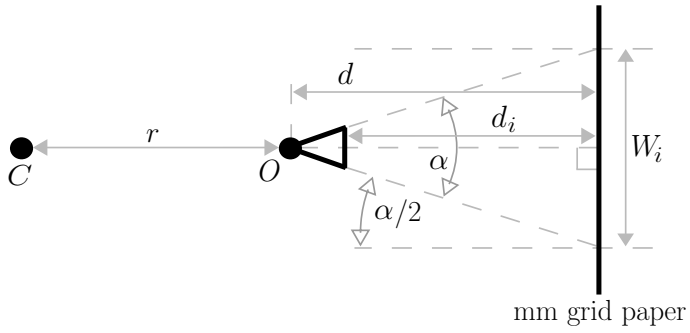


Figure 2.6: The relation between the parameters, which are important for determining the radius r.

r represents the distance between the rotational center of the system and the optical center of the camera. Since the exact position of the optical center is normally not known (not given by the manufacturer), we have to estimate its position. Optical firms with their special equipment would do the best job, but since this has not been an option for us, we have used a simple method, which has been proved quite useful

(Fig. 2.6): First the camera horizontal view angle α has been estimated. Then we have captured a few images of the mm grid paper from known distances d_i from one point on the camera to the paper. The optical axis has been assumed to be perpendicular to the paper surface. From each image we have read the width W_i of it in mm and used all now known values (α, d_i and W_i) to estimate the distance d from the paper to the optical center by manually drawing a geometrically precise relation between the parameters. More distances d_i have been used to check the consistency of all estimates. At the end the position of the optical center has been calculated as an average over all estimated values. Because we know the distances d_i and d, we also know the position of the optical center with respect to the point on the camera from which we have measured the distances d_i. Finally, we can measure the distance r. Nevertheless, this is a rough estimation of the optical center position, but it can be optimized as shown in the experiment in Sec. 2.8.9.

φ determines the column of each captured image, which is mosaiced into the panoramic image. The two models for estimating angle φ (Eqs. (2.6) and (2.8)) differ from one another: the first one is linear, while the second one is not. But since we use cameras with the maximal horizontal view angle $\alpha = 39.72°$, the biggest possible difference between the models is only $0.3137°$ (at the point, where ratio $W_{2\varphi}/W = 91/160$). In the experiments we use such values of φ that the difference is very small, i.e. the biggest difference is lower than $0.1°$. The experiment in Sec. 2.8.6 shows that we obtain slightly better results with the linear model for a given (estimated) set of parameters. This is why the linear model was used in all other experiments.

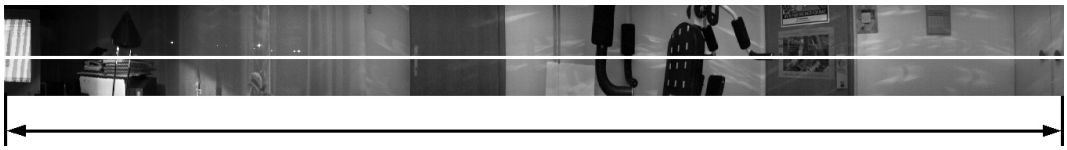
We discuss the angle θ_0 and the radius r in relation with the one-pixel error in estimation of the angle φ in the end of Sec. 2.7.4.

2.7.3 Constraining the search space on the epipolar line

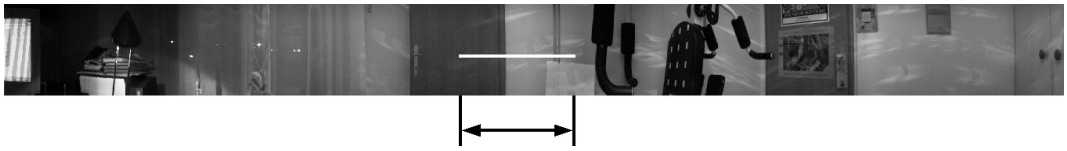
Knowing that the width of the panoramic image is much bigger than the width of the captured image, we would have to search for a corresponding point along a very long epipolar line (Fig. 2.7a). Therefore we would like to constraint the search space on the epipolar line as much as possible. This means that the stereo reconstruction procedure executes faster. A side effect is also an increased confidence in the estimated depth.

From Eq. (2.5) we can derive two conclusions, which nicely constraint the search space:

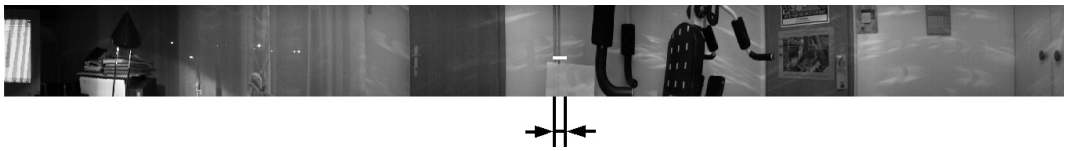
1. Theoretically, the minimal possible estimation of depth is $l_{\min} = r$. This is true for $\theta = 0°$. However, this is impossible in practice since the same point on the scene cannot be seen in the column that will be mosaiced in the panorama for the left eye and at the same time in the column that will be mosaiced in the panorama for the right eye. If we observe the horizontal axis of the panoramic



a) unconstrained length of the epipolar line: 1501 pixels



b) constrained length of the epipolar line: 145 pixels, $2\varphi = 29.9625°$



c) constrained length of the epipolar line: 17 pixels, $2\varphi = 3.6125°$

Figure 2.7: We can effectively constrain the search space on the epipolar line.

image with respects to the direction of the rotation, we can see that every point on the scene that is shown on both panoramic images (Fig. 2.4) is first imaged in the panorama for the left eye and then in the panorama for the right eye. Therefore, we have to wait until the point imaged in the column building up the left eye panorama moves in time to the column building up the right eye panorama. If θ_0 presents the angle by which the camera is shifted, then $2\theta_{\min} = \theta_0$. In consequence, we have to make at least one basic shift of the camera to enable a scene point projected in a right column of the captured image forming the left eye panorama to be seen in the left column of the captured image forming the right eye panorama.

Based on this fact, we can search for the corresponding point in the right eye panorama starting from the horizontal image coordinate $x + \frac{2\theta_{\min}}{\theta_0} = x + 1$ forward, where x is the horizontal image coordinate of the point in the left eye panorama for which we are searching the corresponding point. Thus, we get the value $+1$ since the shift for the angle θ_0 describes the shift of the camera for a single column of the captured image.

In our system, the minimal possible depth estimation $l_{\min}$ depends on the value of the angle φ:

$$\begin{aligned} l_{\min}(2\varphi = 29.9625°) &= 302 \text{ mm} \\ &\dots \\ l_{\min}(2\varphi = 3.6125°) &= 318 \text{ mm}. \end{aligned}$$

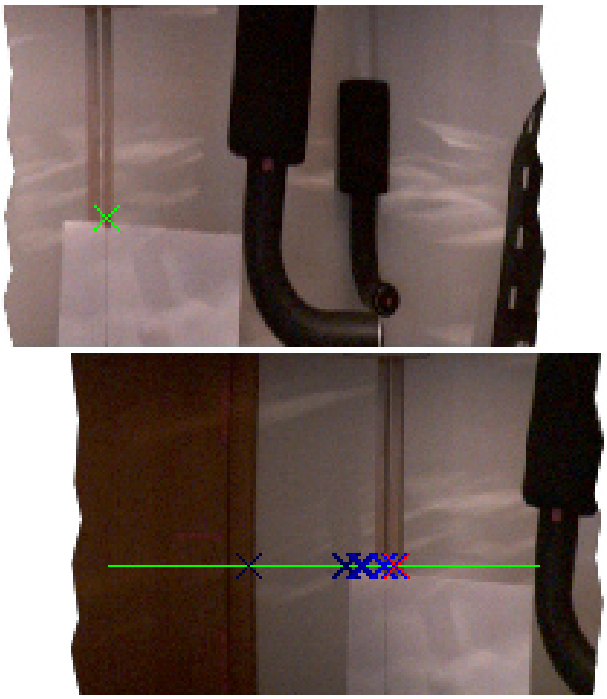


Figure 2.8: Constraining the search space on the epipolar line in case of $2\varphi = 29.9625°$. In the left eye panorama (top image) we have denoted the point for which we are searching the corresponding point with a green cross. In the right eye panorama (bottom image) we have used green color to mark the part of the epipolar line on which the corresponding point must lie. The best corresponding point is marked with a red cross. With blue crosses we have marked a number of points which presented temporary best corresponding point before we actually found the point with the maximal correlation.

2. Theoretically, the estimation of depth is not constrained upwards, but from Eq. (2.5) it is evident that the denominator must be non-zero. Practically, this means that for the maximal possible depth estimation $l_{\max}$ the difference $\varphi - \theta_{\max}$ must be equal to the value in the interval $(0, \frac{\theta_0}{2})$. We can write this fact as: $\theta_{\max} = n \cdot \frac{\theta_0}{2}$, where $n = \varphi \text{ div } \frac{\theta_0}{2}$ and $\varphi \bmod \frac{\theta_0}{2} \neq 0$.

   If we write the constraint for the last point, which can be a corresponding point on the epipolar line, in analogy with the case of determining the starting point that can be a corresponding point on the epipolar line, we have to search for the corresponding point in the right eye panorama to including the horizontal image coordinate $x + \frac{2\theta_{\max}}{\theta_0} = x + n$. Here x is the horizontal image coordinate of the point on the left eye panorama for which we are searching the corresponding point.

Equivalently, like in case of the minimal possible depth estimation $l_{\min}$, the maximal possible depth estimation $l_{\max}$ also depends upon the value of the angle φ:

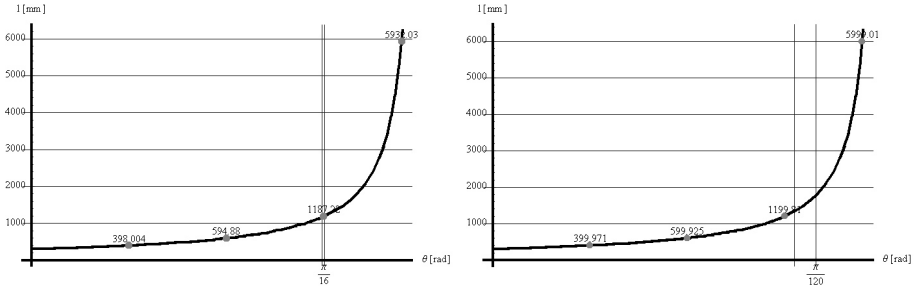
$$\begin{aligned} l_{\max}(2\varphi = 29.9625^\circ) &= 54687 \text{ mm} \\ &\dots \\ l_{\max}(2\varphi = 3.6125^\circ) &= 86686 \text{ mm}. \end{aligned}$$

In the following sections we show that we cannot trust the depth estimates near the last point of the epipolar line search space, but we have proven that we can effectively constrain the search space.

To illustrate the use of specified constraints on real data, let us present the following example which describes the working process of our system: while the width of the panorama is 1501 pixels, when searching for a corresponding point, we have to check only φ div $\frac{\theta_0}{2} = 145$ pixels in case of $2\varphi = 29.9625^\circ$ (Figs. 2.7b and 2.8) and only 17 in case of $2\varphi = 3.6125^\circ$ (Fig. 2.7c).

From the last paragraph we could conclude that the stereo reconstruction procedure is much faster for a smaller angle φ. However, in the next section we show that a smaller angle φ, unfortunately, has also a negative property.

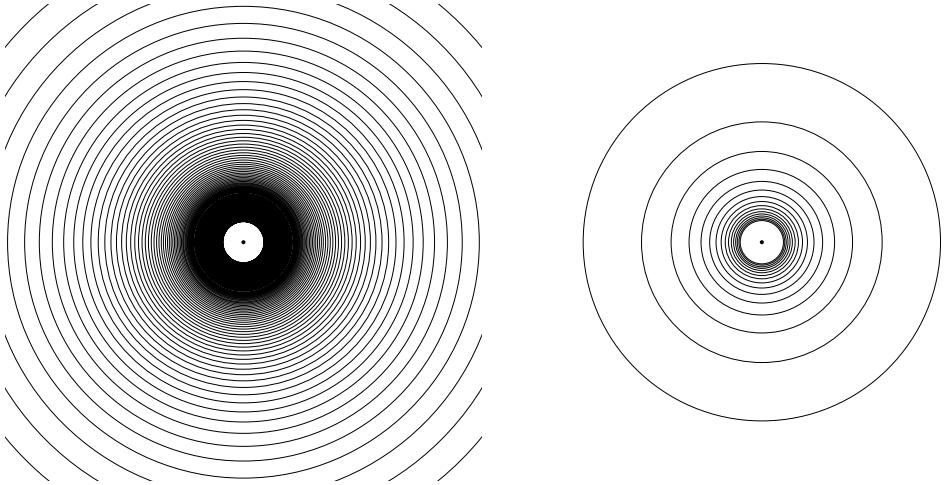
2.7.4 Meaning of the one-pixel error in estimation of the angle θ



a) $2\varphi = 29.9625^\circ$ b) $2\varphi = 3.6125^\circ$

Figure 2.9: The dependence of depth l on angle θ (Eq. (2.5), $r = 30$ cm and two different values of φ are used). To visualize the one-pixel error in estimation of the angle θ, we have marked the interval of width $\frac{\theta_0}{2} = 0.102857^\circ$ between the vertical lines near the third point.

Let us first define what we mean under the term one-pixel error. As the images are discrete, we would like to know what is the value of the error in the depth estimation if we miss the right corresponding point for only one pixel. And we would like to have this information for various values of the angle φ.



a)
$2\varphi = 29.9625°$
145 possible depth estimations

b)
$2\varphi = 3.6125°$
17 possible depth estimations

Figure 2.10: The number of possible depth estimates is proportional to the angle φ. Each circle denotes a possible depth estimation value.

Before we illustrate the meaning of the one-pixel error in estimation of the angle θ, let us take a look at the graphs in Fig. 2.9. The graphs show the dependence of the depth function l on the angle θ when two different values of the angle φ are used. It is evident that the depth function l rises slower in case of a bigger angle φ. This property decreases the error in the depth estimation l when a bigger angle φ is used and this decrease in the error becomes even more evident if we know that the horizontal axis is discrete and the intervals on the axis are $\frac{\theta_0}{2}$ degrees wide (see Fig. 2.9). If we compare the width of the interval in both graphs with respect to the width of the interval that θ is defined in ($\theta \in [0, \varphi]$), we can see that the interval with the width of $\frac{\theta_0}{2}$ degrees is much smaller when a bigger angle φ is used. This subsequently means that the one-pixel error in estimation of the angle θ is much smaller when a bigger angle φ is used, since a shift for the angle θ_0 describes the shift of the camera for a single column of pixels.

Because of a discrete horizontal axis θ (Fig. 2.9), with intervals $\frac{\theta_0}{2}$ degrees wide (in our case $\theta_0 = 0.205714°$), the number of possible depth estimates is proportional to the angle φ: we can calculate φ div $\frac{\theta_0}{2} = 145$ different depth values (Eq. (2.5)) if we use the angle $2\varphi = 29.9625°$ (Fig. 2.10a) and only 17 different depth values if we use the angle $2\varphi = 3.6125°$ (Fig. 2.10b). This is the disadvantage of small angles φ (see the experiment in Sec. 2.8.1).

Let us illustrate the meaning of the one-pixel error in estimation of the angle θ: We would like to know what is the error of the angle θ at the beginning of the

	$\theta - \frac{\theta_0}{2}$	θ	$\theta + \frac{\theta_0}{2}$
l [mm]	394.5	398	401.5
Δl [mm]	3.5		
(error)		3.5	

a) $\theta = \theta_1 = \frac{\varphi}{4}$, $2\varphi = 29.9625°$

	$\theta - \frac{\theta_0}{2}$	θ	$\theta + \frac{\theta_0}{2}$
l [mm]	372.5	400	431.8
Δl [mm]	27.5		
(error)		31.8	

b) $\theta = \theta_1 = \frac{\varphi}{4}$, $2\varphi = 3.6125°$

	$\theta - \frac{\theta_0}{2}$	θ	$\theta + \frac{\theta_0}{2}$
l [mm]	2252.9	2373.2	2507
Δl [mm]	120.3		
(error)		133.8	

c) $\theta = \theta_2 = \frac{7\varphi}{8}$, $2\varphi = 29.9625°$

	$\theta - \frac{\theta_0}{2}$	θ	$\theta + \frac{\theta_0}{2}$
l [mm]	1663	2399.6	4307.4
Δl [mm]	736.6		
(error)		1907.8	

d) $\theta = \theta_2 = \frac{7\varphi}{8}$, $2\varphi = 3.6125°$

Table 2.2: The one-pixel error Δl in estimation of the angle θ, where $r = 30$ cm (Eq. (2.5)).

interval over which θ is defined ($\theta \in [0, \varphi]$) and what is the error of the angle θ near the end of this interval?

For this purpose we choose angles $\theta_1 = \frac{\varphi}{4}$ and $\theta_2 = \frac{7\varphi}{8}$. We are also interested in the nature of the error for different values of the angle φ. In this example we use our already standard values for the angle φ: $2\varphi = 29.9625°$ and $2\varphi = 3.6125°$. The results in Tab. 2.2 give the values of the one-pixel error in estimation of the angle θ for different values of parameters θ and φ.

From the results in Tab. 2.2 we can conclude that the error is much bigger in case of a smaller angle φ than in case of a bigger angle φ. The second conclusion is that the value of the error increases as the value of the angle θ gets closer to the value of the angle φ. This is true regardless of the value of the angle φ. This two conclusions are also evident from Fig. 2.10: possible depth estimations lie on concentric circles centered in the center of the system, with the distance between circles increasing the further away they lie from the center (see also the experiment in Sec. 2.8.3). The figure nicely illustrates the fact that in case of a small angle φ, we can estimate only a few different depths and the fact that the one-pixel error in estimation of the angle θ increases if we move away from the center of the system.

We would like to get reliable depth estimates, but at the same time we would like the reconstruction procedure to execute fast. Here, we are faced with two contradicting requirements, since we have to make a compromise between the accuracy of the system and the speed of the reconstruction procedure. Namely, if we wanted to achieve the maximal possible accuracy, then we would use the maximal possible angle φ. But this means that we would have to conduct a search for the corresponding points on a larger segment of the epipolar line. Consequently, the speed of the reconstruction process would be lower. We would come to the same conclusion if we

wanted to achieve a higher speed of the reconstruction procedure, since the speed of the reconstruction process is inversely proportional to its accuracy.

By varying the parameters θ_0 and r we change the size of the error:

- By increasing the resolution of captured images, we decrease the angle θ_0 (Eq. (2.10)) and subsequently decrease the rotational angle of the camera between two successively captured images forming the stereo panoramic images. By nearly the same factor that we increase (decrease) the resolution of captured images, we decrease (increase) the value of the error Δl, while the reconstruction process takes more (less) time by nearly the same factor. By decreasing (increasing) the value θ_0 we are able to calculate more (less) depth values and consequently, we achieve bigger (lower) accuracy. Another way to influence the parameter θ_0 is to vary the horizontal view angle α. This influence is presented separately in Sec. 2.7.7.

- By the same factor that we increase (decrease) the radius r, we increase (decrease) the (biggest possible and sensible) depth estimation l and the size of error Δl. Obviously, if the camera optical center is at the same distance from one really close object for different r, we achieve bigger accuracy by using smaller r. The behavior of $\Delta l_{\min}$ given in the next section nicely illustrates this fact. If we vary the parameter r, the process of reconstruction is not any faster or slower. In practice, a bigger r means that we can reconstruct bigger scenes (rooms). The geometry of our system is adequate of reconstructing (smaller) rooms and is not really suitable for reconstruction of an outdoor scene. This is due to the inherent property of the system: we do not trust in the estimated depth l of far-away objects on the scene if the size of the error Δl is too big. If we vary the parameter r, the number of possible depth estimates naturally stays the same.

2.7.5 Definition of the maximal reliable depth value

In Sec. 2.7.3 we have defined the minimal possible depth estimation $l_{\min}$ and the maximal possible depth estimation $l_{\max}$, but we have not said anything about the meaning of the one-pixel error in estimation of the angle θ for these two estimated depths. Let us examine the size of the error Δl for these two estimated depths. We calculate $\Delta l_{\min}$ as the absolute value of difference between the depth $l_{\min}$ and the depth l for which the angle θ is bigger than the angle $\theta_{\min}$ by the angle $\frac{\theta_0}{2}$:

$$\Delta l_{\min} = |l_{\min}(\theta_{\min}) - l(\theta_{\min} + \frac{\theta_0}{2})| = |l_{\min}(\frac{\theta_0}{2}) - l(\theta_0)|.$$

Similarly, we calculate the error $\Delta l_{\max}$ as the absolute value of difference between the depth $l_{\max}$ and the depth l for which the angle θ is smaller than the angle $\theta_{\max}$ by the angle $\frac{\theta_0}{2}$:

$$\Delta l_{\max} = |l_{\max}(\theta_{\max}) - l(\theta_{\max} - \frac{\theta_0}{2})| = |l_{\max}(n\frac{\theta_0}{2}) - l((n-1)\frac{\theta_0}{2})|,$$

where the variable n denotes a positive number in equation: $n = \varphi$ div $\frac{\theta_0}{2}$.

	$2\varphi = 29.9625°$	$2\varphi = 3.6125°$
$\Delta l_{\min}$	2 mm	19 mm
$\Delta l_{\max}$	30172 mm	81587 mm

Table 2.3: The one-pixel error Δl in estimation of the angle θ for the minimal possible depth estimation $l_{\min}$ and the maximal possible depth estimation $l_{\max}$ with respect to the angle φ and the radius r=30 cm.

In Tab. 2.3 we have gathered the error sizes for different values of the angle φ. The results confirm statements in Sec. 2.7.4. We can add one additional conclusion: The value of error $\Delta l_{\max}$ is unacceptably high and this is true regardless of the value of the angle φ. This is why we have to sensibly decrease the maximal possible depth estimation $l_{\max}$. In practice, this leads us to defining the upper boundary of the allowed error size (Δl) for a single pixel in the estimation of the angle θ. Using it, we subsequently define the maximal reliable depth value (see the example in the next section).

2.7.6 Contribution of the vertical reconstruction

Addressing the vertical reconstruction is essential for getting as accurate results as possible. In this section we investigate how big is the difference between the depths estimated without (Eq. (2.5)) and with (Eq. (2.9)) addressing the vertical reconstruction.

Let us first define the maximal reliable depth value $l_{\max}$ as suggested in the previous section for the camera with the horizontal view angle $\alpha = 34°$ and the vertical view angle $\beta = 25°$. If we do not allow the error size Δl to be more than 10 cm for $r = 30$ cm, $2\varphi = 29.9625°$ and $\theta_0 = 0.205714°$, then, consequently, $l_{\max} = 213.5$ cm. By introducing the influence of the vertical view angle β into Eq. (2.9):

$$\omega_{1\ \max} = \frac{\beta}{2},$$

we get $l_{\max}(\alpha, \beta) = 217.4$ cm (Fig. 2.11). This means that the contribution of the vertical reconstruction is small ($l_{\max}(\alpha, \beta) - l_{\max} = 3.9$ cm, which is 1.8% of $l_{\max}(\alpha, \beta)$), but as expected it has a positive influence on the overall results as shown in the experiment in Sec. 2.8.4. By increasing (decreasing) the angle β (using different cameras) we also increase (decrease) the contribution of the vertical reconstruction.

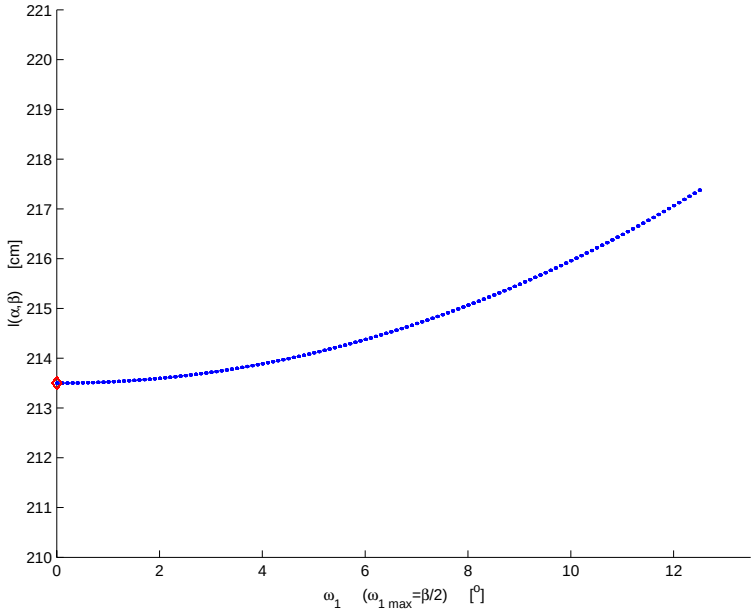


Figure 2.11: The contribution of the vertical reconstruction is small for the camera with the horizontal view angle $\alpha = 34°$ and the vertical view angle $\beta = 25°$ (Eq. (2.9)). The diamond marks the depth $l_{\max}$ estimated without addressing the vertical reconstruction (Eq. (2.5)). For detailed description see Sec. 2.7.6.

2.7.7 Influence of using different cameras

Each camera can be characterized by its horizontal view angle α. According to Eq. (2.10), the angle θ_0 gets bigger with bigger α, having in mind that the width W (the resolution) of the captured images stays the same. This means that we have to capture less images with the camera characterized by bigger α in order to generate the panoramic image of the same scene. Let us illustrate this fact by presenting generated panoramic images of the same scene, where we varied α (we used different cameras). In Fig. 2.12 we can see that for the cameras mentioned in the end of Sec. 2.4, we get panoramic images of different horizontal resolution, while the vertical resolution is equal for all cameras:

a) the camera with the horizontal view angle $\alpha = 16.53°$ gives a panoramic image with the width $W_{pan} = 3001$ pixels,

b) the camera with the horizontal view angle $\alpha = 34°$ gives a panoramic image with the width $W_{pan} = 1501$ pixels and

c) the camera with the horizontal view angle $\alpha = 39.72°$ gives a panoramic image with the width $W_{pan} = 1201$ pixels.

a) $\alpha = 16.53° \Longrightarrow W_{pan} = 3001$ pixels

b) $\alpha = 34° \Longrightarrow W_{pan} = 1501$ pixels

c) $\alpha = 39.72° \Longrightarrow W_{pan} = 1201$ pixels

Figure 2.12: Different cameras characterized by the horizontal view angle α give panoramic images with different horizontal resolution W_{pan}.

So, by enlarging the camera horizontal field of view the width of the panorama gets lower, while the height of the panorama stays the same. But the same height captures more scene, since also the camera vertical field of view is enlarged.

This means that we generate a panoramic image faster if we use a camera with a wider view angle. But the drawback here is that the horizontal angular resolution (the number of possible depth estimates per one degree) gets lower. As already (implicitly) mentioned in the end of Sec. 2.7.4, by varying the resolution of the captured images we vary the horizontal and the vertical resolution of the generated panoramic image at fixed α. But now we can add one more conclusion, namely, if we vary α then we vary only the horizontal resolution of the generated panoramic image at fixed resolution of the captured images.

For each camera (and not only for the cameras that we use) the maximal number of possible depth estimates depends on the horizontal resolution W of captured images. From $\varphi_{\max} = \alpha/2$, θ_0 and the equation for determining the number of possible depth estimates $\varphi_{\max}$ div $\frac{\theta_0}{2}$, we get very similar results for different cameras (α) at fixed W. (All the results are equal to W if $\theta_0(\varepsilon) = \theta_0(\alpha)$ (see the discussion on estimation of the angle θ_0 ($\theta_0(\alpha)$, $\theta_0(\varepsilon)$) in Sec. 2.7.2).) This means that the comparison of results gained using different cameras should not be done at the similar φ, but rather at the similar number of the possible depth estimates. This fact is used in the experiment in Sec. 2.8.8.

The size of the one-pixel error Δl in estimation of the angle θ (Sec. 2.7.4), for φ's defined in described way, is also similar. This is evident from Tab. 2.4.

		θ	$\theta + \frac{\theta_0}{2}$
l [mm]	$2\varphi = 29.9625°$ ($\alpha = 34°$)	398	401.5
	$2\varphi = 38.47875°$ ($\alpha = 39.72°$)	396.7	400.2
Δl [mm] (error)	$2\varphi = 29.9625°$ ($\alpha = 34°$)	3.5	
	$2\varphi = 38.47875°$ ($\alpha = 39.72°$)	3.5	

a) $\theta = \theta_1 = \frac{\varphi}{4}$

		θ	$\theta + \frac{\theta_0}{2}$
l [mm]	$2\varphi = 29.9625°$ ($\alpha = 34°$)	2373.2	2507
	$2\varphi = 38.47875°$ ($\alpha = 39.72°$)	2355.8	2488.8
Δl [mm] (error)	$2\varphi = 29.9625°$ ($\alpha = 34°$)	133.8	
	$2\varphi = 38.47875°$ ($\alpha = 39.72°$)	133	

b) $\theta = \theta_2 = \frac{7\varphi}{8}$

Table 2.4: The one-pixel error Δl in estimation of the angle θ for different cameras (α) at the similar number of the possible depth estimates following from φ, where $r = 30$ cm (Eq. (2.5)). For $2\varphi = 29.9625°$ the results are the same as in Tab. 2.2. Consequently, the same θ values are used in this table.

2.8 Experimental results

In the experiments the following cameras were used:

- *camera #1* with parameters:
  - $\alpha = 34°$
  - $\beta = 25°$
  - $r = 30$ cm
  - $2\varphi = 29.9625°$
  - $\theta_0 = 0.205714°$
- *camera #2* with parameters:
  - $\alpha = 39.72°$
  - $\beta = 30.54°$
  - $r = 31$ cm
  - $2\varphi = 38.47875°$
  - $\theta_0 = 0.257143°$
- *camera #3* with parameters:
  - $\alpha = 16.53°$
  - $\beta = 12.55°$
  - $r = 35.6$ cm
  - $2\varphi = 15.3935625°$
  - $\theta_0 = 0.102857°$.

All the panoramic images are generated from images with resolution of 160×120 pixels.

Correspondences for each feature point on the scene used in the evaluation have been determined with a *normalized correlation* procedure [4] and rechecked manually for consistency. If the difference between the manually and the automatically determined correspondence has been more than one pixel, such feature has not been used in the evaluation, otherwise we believe in the automatically obtained result rather than in the manually obtained result, because the latter is a subjective result, while the other is an objective result. Fact is that it is hard to manually determine the corresponding point due to the discrete nature of images. Nevertheless, in more than 75% the two results have been the same.

We use the normalized correlation procedure to search for corresponding points because it is one of the most commonly used technique employed for that purpose.

On the other hand, correlation-based stereo algorithms are the only ones that can produce sufficiently dense depth images with an algorithmic structure which lends itself nicely to fast implementations because of the simplicity of the underlying computation [5]. Various improvements to real time correlation-based stereo vision are discussed in [5, 45]. To improve the results we could also employ multiple-baseline approach [6, 36]. It has been shown that by using multiple-baseline stereo, match ambiguities can be reduced and the reconstruction precision can be improved as well. Other interesting methods than just those based on correlation are described in [2]. A nice survey about a taxonomy and evaluation of dense two-frame stereo correspondence algorithms is given in [48]. In [55], the authors review recent advances in computational stereo, focusing primarly on correspondence methods, methods for occlusion and real time implementations.

The normalized correlation procedure uses the principle of similarity of scene parts within two scene images. The basic idea of the procedure is to find the part of the scene in the second image which is most similar to a given part of the scene in the first image. The procedure uses a window, within which the similarity is measured with help of the correlation technique.

We use this procedure also when we generate depth images. Additionally, to increase the confidence in the estimated depth, we employ a procedure called *back-correlation* [4]. The main idea of this procedure is to first find a point $\mathbf{m}_2$ in the second image which corresponds to a point $\mathbf{m}_1$ given in the first image. Then we have to find the point corresponding to the point $\mathbf{m}_2$ in the first image. Let us denote this corresponding point by $\mathbf{m}_1'$. If the point $\mathbf{m}_1$ is equal to the point $\mathbf{m}_1'$ then we keep the estimated depth value. Otherwise, we do not keep the estimated depth value. This means that the point $\mathbf{m}_1$, for which the back-correlation was not successful, has no depth estimation associated with it in the depth image. Using the back-correlation procedure we also solve the problem of occlusions. On the other hand, the normalized correlation score can also be used for estimating the confidence in the estimated depth.

All results were generated by using a correlation window of size $2n+1 \times 2n+1$, n=4, if not mentioned otherwise. We searched for corresponding points only in the panoramic image row determined by the epipolar geometry.

The primary evaluation of the system is based on mentioned feature points on the scene. The quantitative measure, which gives the average error of the estimated depth (l (Eq. (2.5)) or $l(\alpha, \beta)$ (Eq. (2.9))) in comparison to the actual distance (d) over n scene points, is calculated as:

$$AVG_{\%} = \frac{\sum_{i=1}^{n} |l_i - d_i|/d_i}{n} \cdot 100\%.$$

The second measure, which is in the results written right beside the first one, is the standard deviation following from:

$$SD_{\%} = \sqrt{\frac{\sum_{i=1}^{n} \left(\frac{|l_i - d_i|}{d_i} \cdot 100\% - AVG_{\%} \right)^2}{n-1}},$$

which reveals how tightly all the various estimated depths are clustered around the average error in the set of data.

On the other hand, the evaluation is also given qualitatively, i.e. visually, where this is needed.

Note that all the presented results are rounded upon their calculations and not before.

Every time we refer to the features on the scene in tables or figures, the appropriate features are also marked for better orientation in the panoramic image given at the bottom of tables and figures.

In the first three experiments (Secs. 2.8.1, 2.8.2 and 2.8.3) we use l (Eq. (2.5)) rather than $l(\alpha, \beta)$ (Eq. (2.9)), so that afterwards we are able to demonstrate the influence of the vertical reconstruction on the reconstruction accuracy.

2.8.1 Influence of different φ values on the reconstruction accuracy — The quantitative evaluation

Experiment background: See the discussion on the number of possible depth estimates with respect to the angle φ in Sec. 2.7.4. The results were obtained with *camera #1*.
Results: The comparison of results using $2\varphi = 3.6125°$ and $2\varphi = 29.9625°$ (see Sec. 2.7.1 about how these values were obtained) is presented in Tab. 2.5.

feature	d [cm]	$2\varphi = 3.6125°$ l [cm]	$2\varphi = 3.6125°$ $l-d$ [cm (% of d)]	$2\varphi = 29.9625°$ l [cm]	$2\varphi = 29.9625°$ $l-d$ [cm (% of d)]
1	111.5	89.4	-22.1 (-19.8%)	109	-2.5 (-2.3%)
2	95.5	76.7	-18.8 (-19.6%)	89.3	-6.2 (-6.5%)
3	64	53.8	-10.2 (-15.9%)	59.6	-4.4 (-6.9%)
4	83.5	76.7	-6.8 (-8.1%)	78.3	-5.2 (-6.2%)
5	92	89.4	-2.6 (-2.8%)	89.3	-2.7 (-2.9%)
6	86.5	76.7	-9.8 (-11.3%)	82.7	-3.8 (-4.4%)
7	153	133.4	-19.6 (-12.8%)	159.8	6.8 (4.5%)
8	130.5	133.4	2.9 (2.2%)	135.5	5 (3.8%)
9	88	76.7	-11.3(-12.8%)	87.6	-0.4 (-0.5%)
10	92	89.4	-2.6 (-2.8%)	89.3	-2.7 (-2.9%)
11	234.5	176.9	-57.6 (-24.6%)	213.5	-21 (-8.9%)
12	198	176.9	-21.1 (-10.7%)	179.1	-18.9 (-9.5%)
13	177	176.9	-0.1 (-0.1%)	186.7	9.7 (5.5%)
		$AVG_{\%}$=11% ± 7.7%		$AVG_{\%}$=5% ± 2.6%	

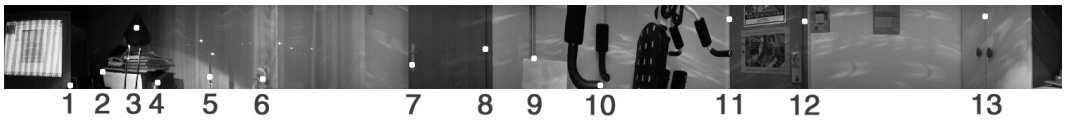


Table 2.5: The comparison of results for two different values of φ.

Conclusion: As expected, the results with $2\varphi = 29.9625°$ are much better, since this angle ensures many more possible depth estimates.

2.8.2 Time analysis of the stereo reconstruction process

Experiment background: Searching for the corresponding point presents the most expensive part of the stereo reconstruction process. In this section we present some time results, given in hours, minutes and seconds, though the ratios between these results are more important, since the measured times depend on the code itself (optimized or unoptimized, sequential or parallel processing), the stereo-matching algorithm, the speed of the processor, the number of processors etc. As already mentioned, our code is not optimized, no processing is done in parallel, we use normalized correlation algorithm and all the calculations are done on a 350 MHz Intel PII PC (in C++ programming language). For better illustration we have run the reconstruction process over the whole generated pair of stereo panoramic images.

On one side, we have constructed dense panoramic images, which means that we have tried to find the corresponding point in the right eye panorama for every point in the left eye panorama.

On the other side, the sparse depth images have been created by searching only for the correspondences of feature points in input panoramic images. The feature points used have been vertical edges on the scene, derived by filtering the panoramic images with the Sobel filter for searching the vertical edges [1, 4]. The time needed for locating the features on the scene reconstructed in the sparse depth image is included in the presented times. But the time needed for acquisition of panoramic images is not included in the reconstruction time.

Some of the generated depth images are presented in the next section.

The results were obtained with *camera #1*.

Results: The comparison of results using $2\varphi = 3.6125°$, $2\varphi = 29.9625°$ and the back-correlation algorithm ($BC = true$ or $false$), while building dense and sparse depth images, is given in Tab. 2.6.

	sparse depth image reconstruction time [min./sec.]	dense depth image reconstruction time [hours/min./sec.]
$2\varphi = 29.9625°$ $BC = true$	1/10	6/42/20
$2\varphi = 29.9625°$ $BC = false$	0/38	3/21/56
$2\varphi = 3.6125°$ $BC = true$	0/33	0/52/56
$2\varphi = 3.6125°$ $BC = false$	0/21	0/29/6

Table 2.6: The comparison of the stereo reconstruction times.

Conclusion: As expected, the time needed for the reconstruction with the back-correlation search is approximately twice the time needed for the reconstruction without it, while the back-correspondence search algorithm has the same complexity as the correspondence search algorithm (because the basic algorithm is the same in both cases, just the role of the stereo images are swapped). And if we use the smaller angle φ, the reconstruction times are up to approximately eight times smaller from presented ones. This is due to the fact that in case of smaller angle φ we have to check only 17 pixels on the epipolar line, while in case of bigger angle φ we have to check 145 pixels on the epipolar line. The ratio between these two numbers is approximately equal to the speed-up factor.

As mentioned, all results have been generated by using a correlation window of size $2n+1 \times 2n+1$, n=4. For comparison, if n=3 then the time needed to create the dense panoramic depth image, while $2\varphi = 29.9625°$ and $BC = true$, is 4 hours, 20 minutes and 55 seconds. The ratio between the window areas is again approximately equal to the speed-up factor. On the other hand, if we run the same process on the faster computer (PC Intel PIV/2.0 GHz), the time needed to gain the same result is 1 hour, 1 minute and 29 seconds. The speed-up factor could again be attributed to the ratio between the processor frequencies. Nevertheless, the newer processor is approximately 4 times faster, which means that after optimizing the code, introducing Intel's MMX SIMD (Single Instruction Multiple Data) instruction set [15, 55] etc., we would gain the sparse panoramic depth image for $2\varphi = 29.9625°$ and $BC = true$ in real time. At this point, real time to us means one stereo reconstruction per second. For autonomous robot navigation the sparse depth image based on vertical edges already contains important information about the environment.

Further stereo reconstruction process speed-up could be achieved by processing 8-bit grayscale images with lower resolution, by doing the reconstruction of only part of the scene in which we are interested, using the property of successive pixels in the panoramic images to constrain the search space on the epipolar line even more, using different stereo-matching algorithm etc. But the most efficient way to ensure the real time reconstruction (at video rate) is to employ cluster of computers, doing real parallel processing [57]. Until very recently, all truly real time implementations made use of special purpose hardware, like digital signal processors (DSP) or field programmable gate arrays (FPGA) [5, 55].

On the other hand, real time correlation based stereo algorithms are discussed in [5, 45, 60]. In the latter, i.e. [60], the real time dense reconstruction is performed on symmetric multiperspective panoramic images with resolution of 1324×120 pixels. The reconstruction is done in 0.34 seconds on a 1.7 GHz PC.

According to Sec. 2.7.7, the speed-up could also be achieved if we use a camera with a wider field of view, since this means that the width of the generated panoramic images is lower. Consequently, the speed of the reconstruction process is higher. If we generate the sparse depth image then the speed-up is not that noticeable, since the number of pixels presenting edges is more or less the same. But in case of dense

depth image the speed-up factor can be substantial: The basic speed-up factor is given by the ratio between the widths of panoramic images.

Real time, on the other hand, is a wide term, as it has different meanings in relation with different applications and consequently, in our case, with demanded reconstruction accuracy.

Let us at the end of this section also touch the storage requirements. Our panoramic images are each of approximately 0.5 MB in size (bmp format), while in [43] the size of each panoramic image is approximately 3400 MB (format is not specified). Their images are really of hyper-resolution (19478×5184 pixels), but acquisition requirements (time, storage, processing, cost) are obviously of great pretension.

2.8.3 Influence of different φ values on the reconstruction accuracy — The qualitative evaluation

Experiment background: We have used a simple stereo-matching algorithm based on the correlation technique. In spite of that, we are interested in how good the obtained results, i.e. depth images, are visually. Since it is hard to evaluate the quality of generated depth images, we present four reconstructions of the room from generated depth images. In this way, we are able to evaluate the quality of generated depth images and consequently the quality of the system. The plan of the room that we have reconstructed is given in Fig. 2.13. In the sketch we have marked the features on the scene that help us evaluate the quality of generated depth images. The result of the (3D) reconstruction process is a ground-plan of the scene. The goal of the experiment is to see how well the reconstruction fits the real room. The results were obtained with *camera #1.*

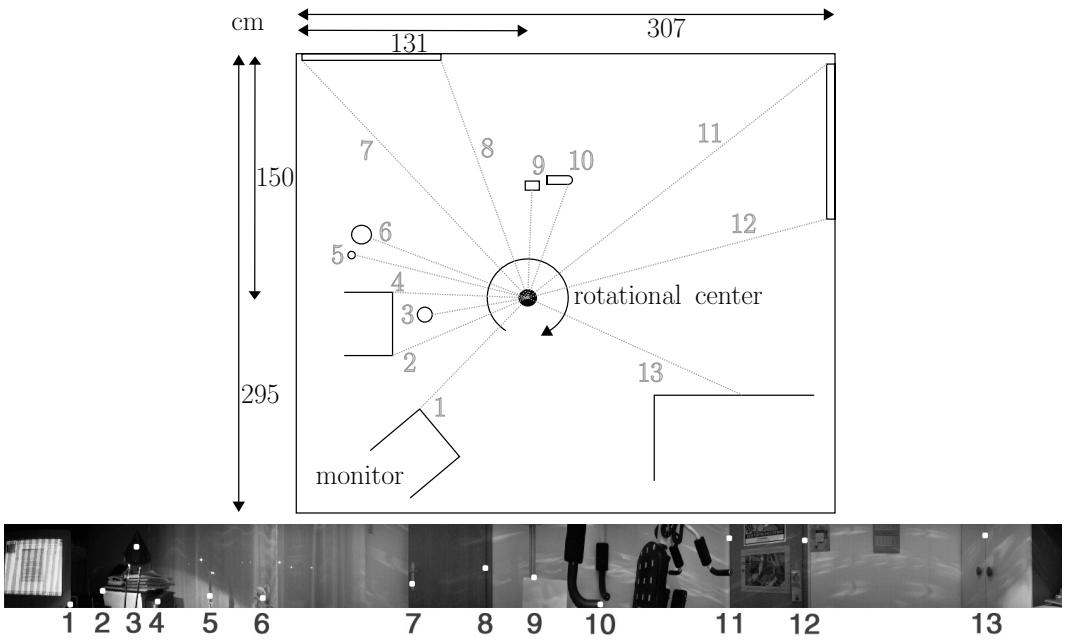
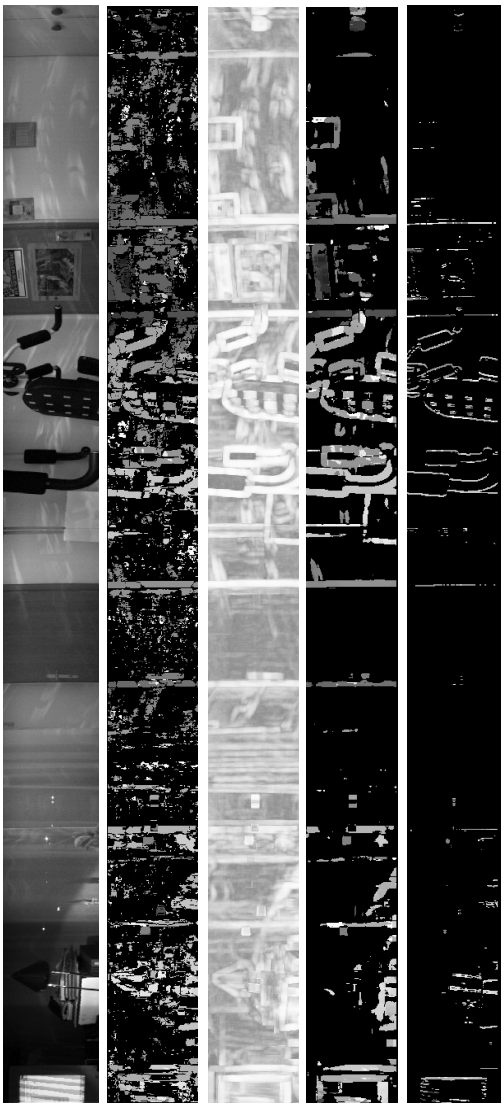


Figure 2.13: The top picture contains the plan of the reconstructed room. In the bottom picture we have marked the features on the scene that help us evaluate the quality of generated depth images.

Results: Fig. 2.14 shows some results of our system. In case denoted with b), we have constructed the dense panoramic image. Black color marks the points on the scene with no depth estimation associated. Otherwise, the nearer the point on the scene is to the rotational center of the system, the lighter the point appears in the depth image.



a) b) c) d) e)

Figure 2.14: Some stereo reconstruction results when creating the depth image for the left eye at the angle $2\varphi = 29.9625°$: a) the left eye panorama, b) a dense depth image / using back-correlation / reconstruction time: 6 hours, 42 min., 20 sec., c) confidence in the estimated depth, d) the dense depth image after weighting / without back-correlation / reconstruction time: 3 hours, 21 min., 56 sec., e) a sparse depth image / without back-correlation / reconstruction time: 38 seconds. The number of pixels for which we searched for the correspondences in case of b) was 147840 ($\times 2$ due to employed back-correlation) and only 4744 in case of e). In case of b) we calculated 21436800 ($\times 2$) correlation scores and in case of e) only 67110 scores.

In case denoted with d), we have used the information about the confidence in the estimated depth (case c), which we get from the normalized correlation estimations. In this way, we have eliminated from the dense depth image all depth estimates which do not have a high enough associated confidence estimation. The lighter the point appears in case c), the more we trust in the estimation of the normalized correlation for this point. In case marked with e), we have created a sparse depth image by searching only for the correspondences of feature points (vertical edges) in input panoramic images.

The following properties are common to the (3D) reconstructions in Figs. 2.15, 2.16, 2.17 and 2.18:

- Big dots denote the actual positions of features on the scene (measured by hand).
- A big dot near the center of the reconstruction shows the position of the center of our system.
- Small black dots represent reconstructed points on the scene.
- Lines between black dots denote links between two successively reconstructed points.

The result of the reconstruction process based on the 68th row of the dense depth image is given in Fig. 2.15 for the angle $2\varphi = 29.9625°$ and in Fig. 2.16 for the angle $2\varphi = 3.6125°$. We have used back-correlation and weighting. In Figs. 2.15 and 2.16 black dots are reconstructed on the basis of the estimated depth values, which are stored in the same row of the depth image. The features on the scene marked with big dots are not necessarily visible in the same row.

We have built sparse depth images by first detecting vertical edges in panoramic images. We have made an assumption that points on vertical edges have the same depth which is approximately true in the examples shown here. The results of the reconstruction shown in Figs. 2.17 and 2.18 are based on information within the entire sparse depth image: first, we calculate the average depth within each column of the depth image and then we show this average depth value in the ground-plan of the scene. In Figs. 2.17 and 2.18 the results have been derived from the sparse depth image gained by using back-correlation. The result in Fig. 2.17 is given for the angle $2\varphi = 29.9625°$ and the result in Fig. 2.18 is given for the angle $2\varphi = 3.6125°$. We imposed one additional constraint on the reconstruction process: each column in the depth image must contain at least four points with associated depth estimates or the average depth is not shown in the ground-plan of the scene.

Conclusion: Although the correlation technique has been used the presented results are good: We can see that the reconstructions correspond to the outline of the room and that the reconstructions support well the statements made throughout Sec. 2.7 — and this was exactly the point of this experiment.

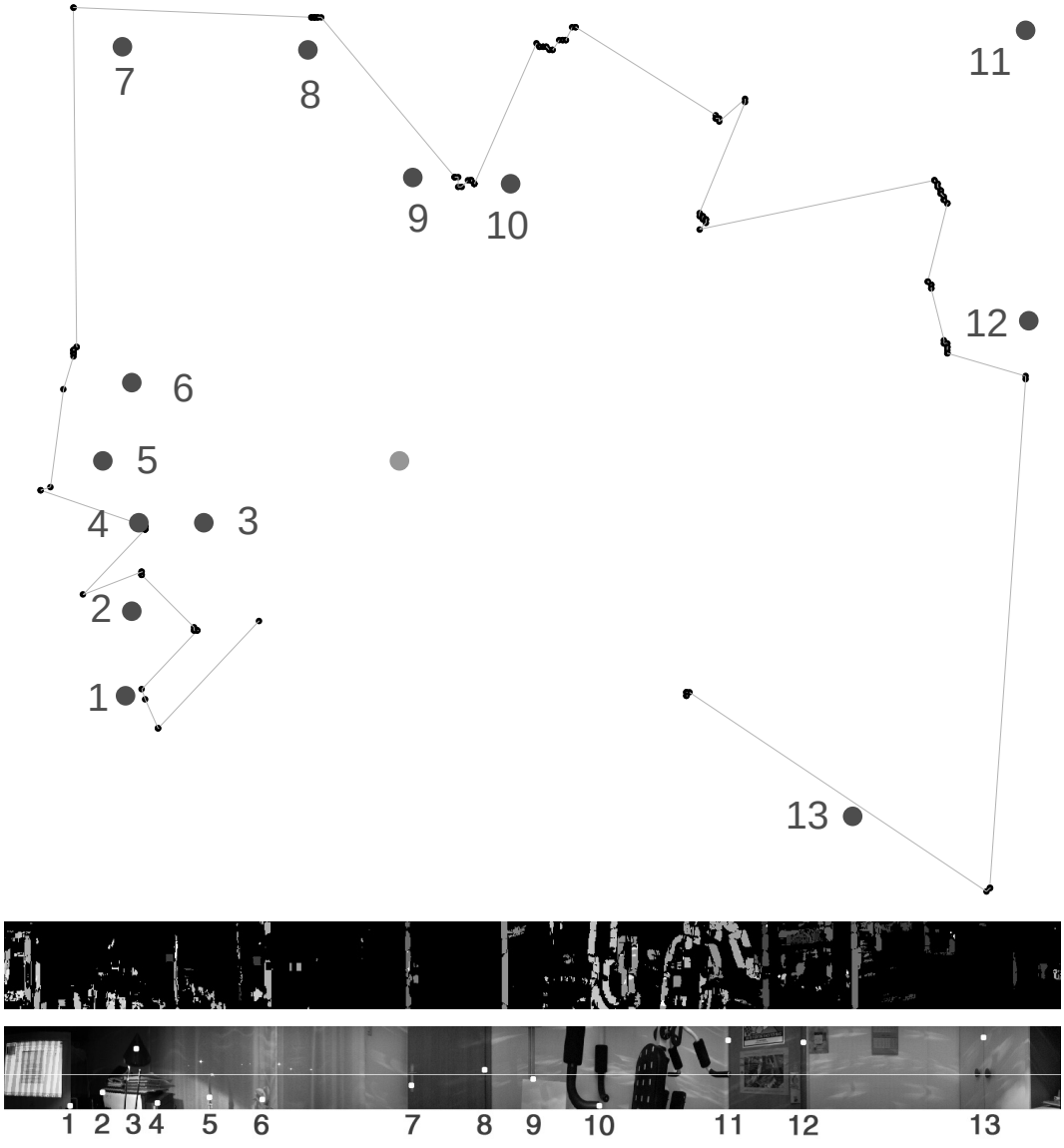


Figure 2.15: The top picture is a ground-plan showing the result of the reconstruction process based on the 68th row of the dense depth image. We have used back-correlation and weighting for the angle $2\varphi = 29.9625°$. The corresponding depth image is shown in the middle picture. For better orientation, the reconstructed row and the features on the scene for which we have measured the actual depth by hand are shown in the bottom picture. The features on the scene marked with big dots and associated numbers are not necessarily visible in this row.

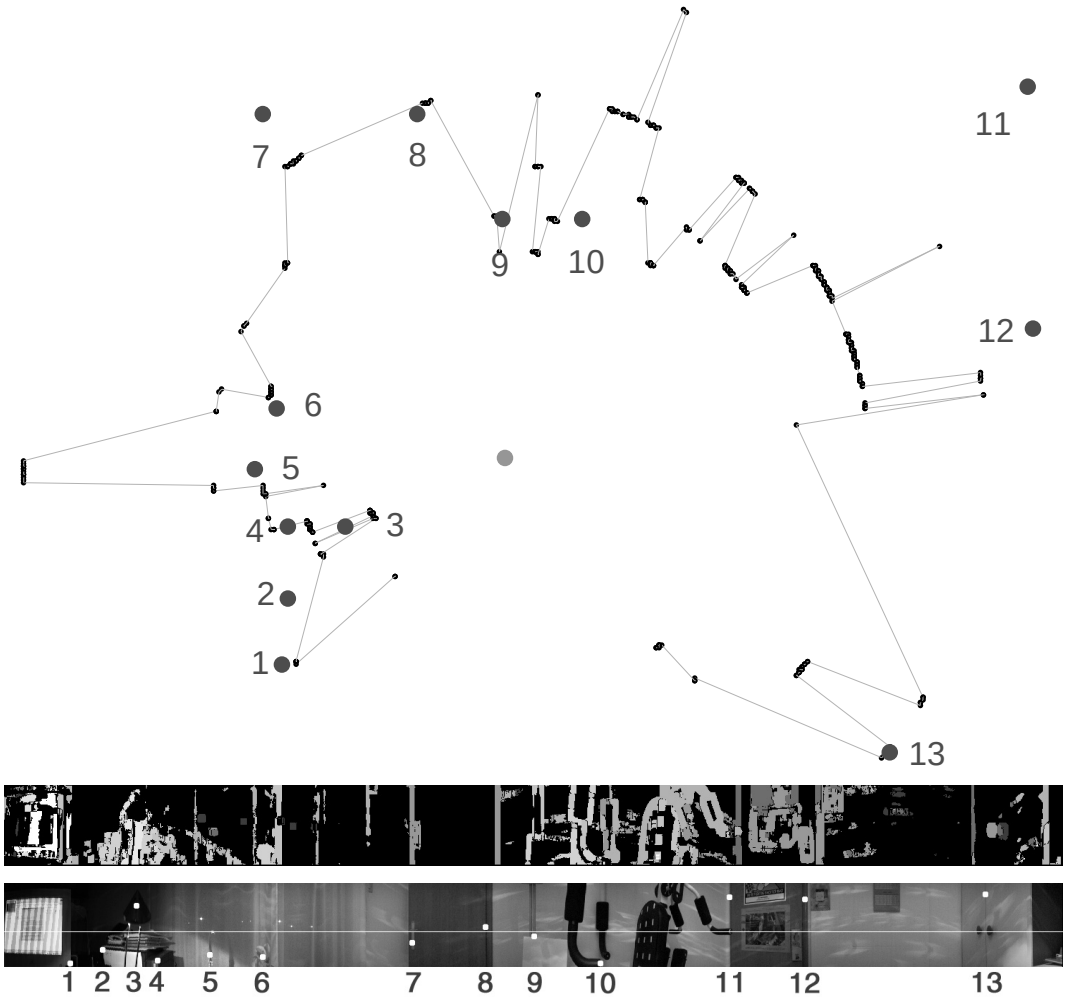


Figure 2.16: The top picture is a ground-plan showing the result of the reconstruction process based on the 68th row of the dense depth image. We have used back-correlation and weighting for the angle $2\varphi = 3.6125°$. The corresponding depth image is shown in the middle picture. For better orientation, the reconstructed row and the features on the scene for which we have measured the actual depth by hand are shown in the bottom picture. The features on the scene marked with big dots and associated numbers are not necessarily visible in this row.

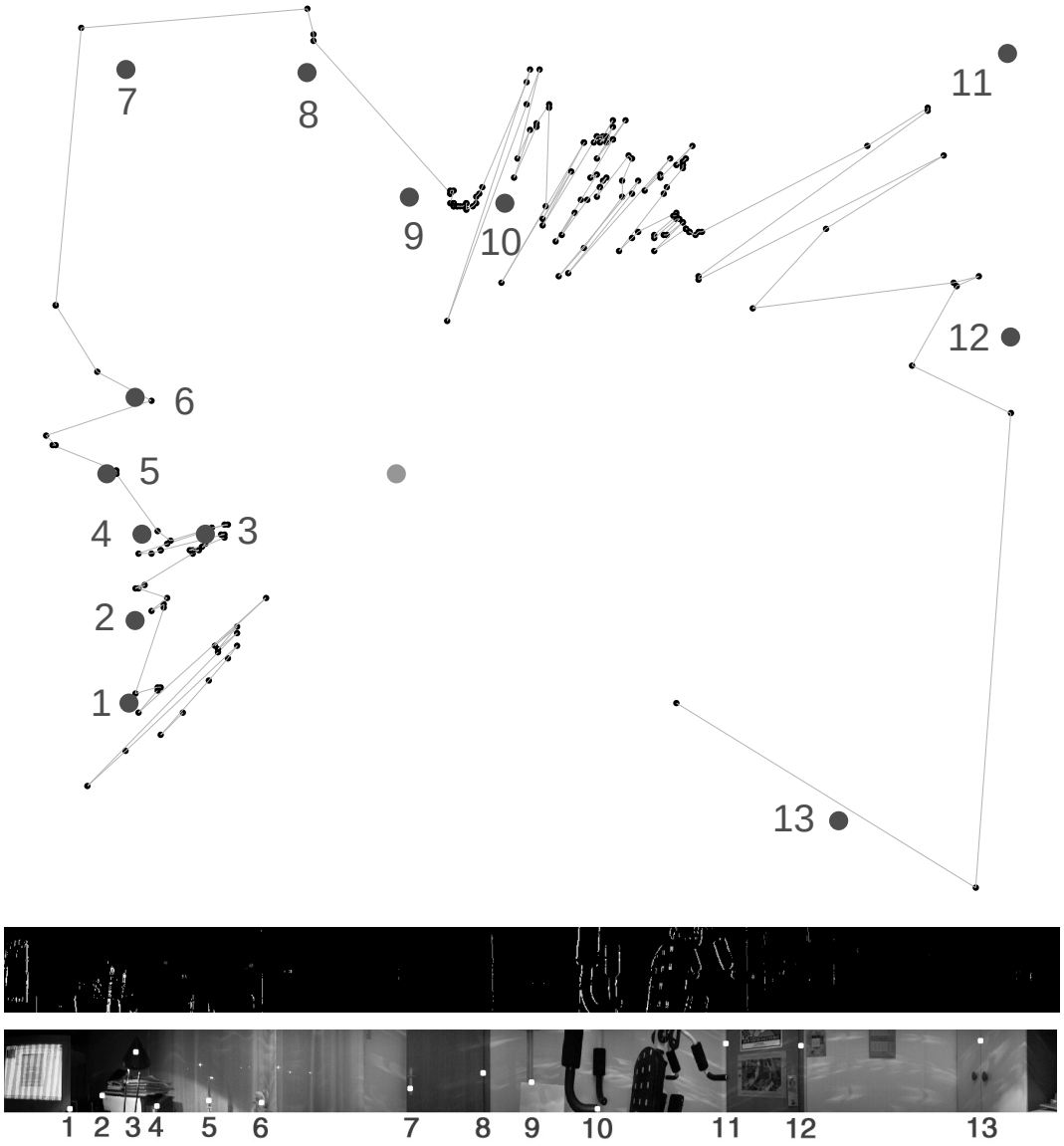


Figure 2.17: The top picture is a ground-plan showing the result of the reconstruction process based on the average depth within each column of the sparse depth image. We have used back-correlation for the angle $2\varphi = 29.9625°$. The corresponding sparse depth image is shown in the middle picture.

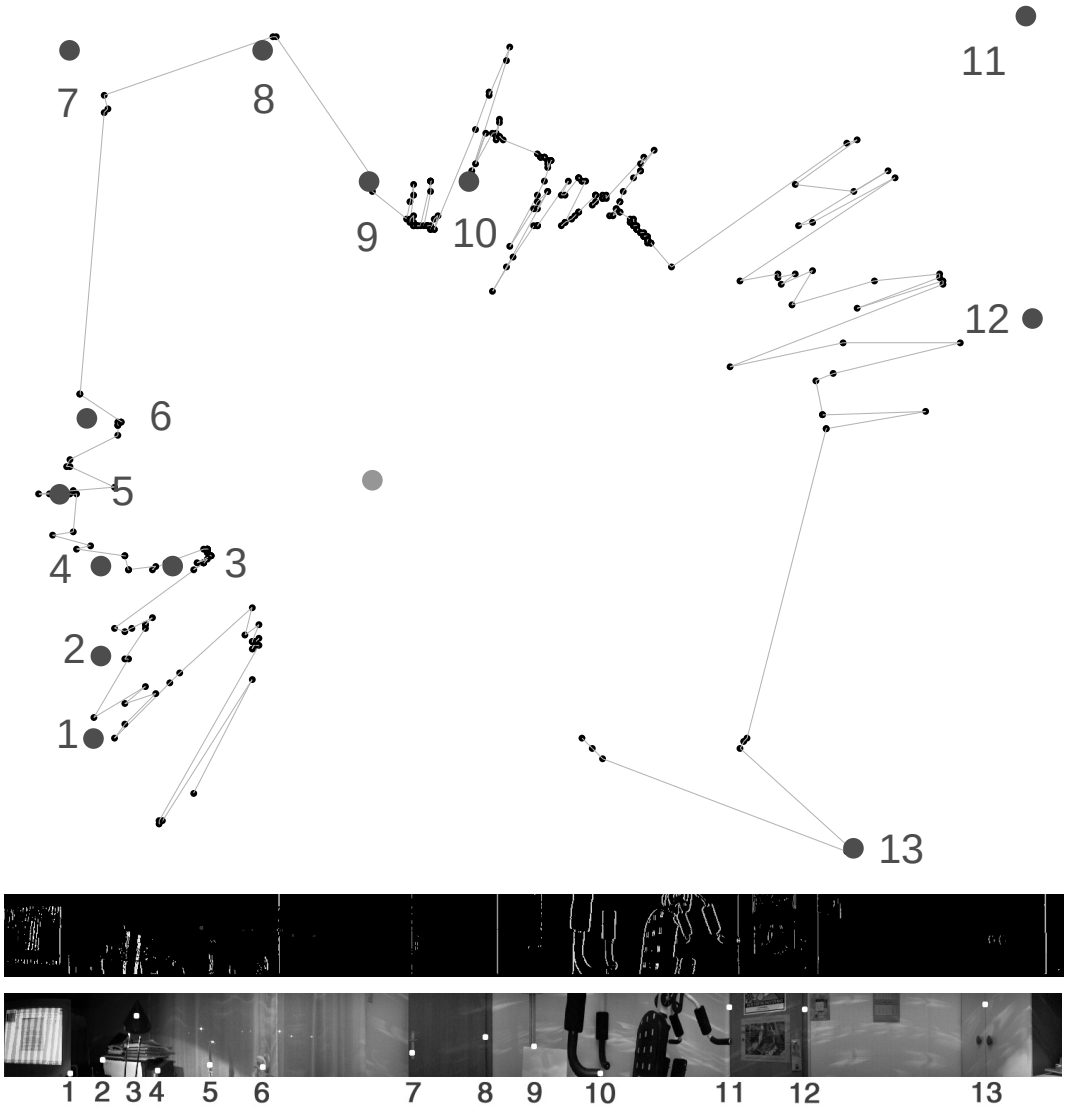


Figure 2.18: The top picture is a ground-plan showing the result of the reconstruction process based on the average depth within each column of the sparse depth image. We have used back-correlation for the angle $2\varphi = 3.6125°$. The corresponding sparse depth image is shown in the middle picture.

In Fig. 2.16 we can observe two properties of the system (Sec. 2.7.4): the reconstructed points are on concentric circles centered in the center of the system and the distance between the circles increases the further away they lie from the center. The figure nicely illustrates the fact that in case of a small angle φ, we can estimate only a few different depths and the fact that the one-pixel error in estimation of the angle θ increases as we move away from the center of the system.

As expected, the correlation technique had performed badly on uniform parts of the scene (e.g. walls), while the edges on the scene are well exposed and assessed in the depth images.

2.8.4 Influence of addressing the vertical reconstruction

Experiment background: In Sec. 2.7.6 we have showed that the contribution of the vertical reconstruction is small. Here, we want to prove its positive influence on the overall accuracy of the system. The results were obtained with *camera #1.*
Results: The comparison of depth estimates l (Eq. (2.5)) and $l(\alpha, \beta)$ (Eq. (2.9)) is presented in Tab. 2.7.

feature	d [cm]	l [cm]	$l-d$ [cm (% of d)]	$l(\alpha,\beta)$ [cm]	$l(\alpha,\beta)-d$ [cm (% of d)]
1	111.5	109	-2.5 (-2.3%)	110.0	-1.5 (-1.3%)
2	95.5	89.3	-6.2 (-6.5%)	89.6	-5.9 (-6.1%)
3	64	59.6	-4.4 (-6.9%)	59.6	-4.4 (-6.8%)
4	83.5	78.3	-5.2 (-6.2%)	78.8	-4.7 (-5.6%)
5	92	89.3	-2.7 (-2.9%)	89.8	-2.2 (-2.4%)
6	86.5	82.7	-3.8 (-4.4%)	83.1	-3.4 (-3.9%)
7	153	159.8	6.8 (4.5%)	160.2	7.2 (4.7%)
8	130.5	135.5	5 (3.8%)	135.5	5.0 (3.8%)
9	88	87.6	-0.4 (-0.5%)	87.6	-0.4 (-0.4%)
10	92	89.3	-2.7 (-2.9%)	90.1	-1.9 (-2.1%)
11	234.5	213.5	-21 (-8.9%)	215.0	-19.5 (-8.3%)
12	198	179.1	-18.9 (-9.5%)	180.4	-17.6 (-8.9%)
13	177	186.7	9.7 (5.5%)	188.5	11.5 (6.5%)
		$AVG_{\%}$=5% ± 2.6%		$AVG_{\%}$=4.7% ± 2.7%	

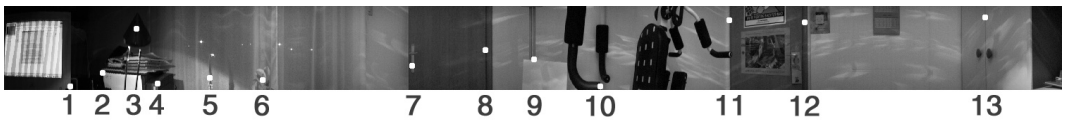


Table 2.7: The comparison of results without and with addressing the vertical reconstruction.

Conclusion: As expected, addressing the vertical reconstruction brings better results. This was observed also in other cases: reconstructions of different rooms using different cameras.

2.8.5 Influence of different θ_0 values on the reconstruction accuracy

Experiment background: See the discussion on estimation of the angle θ_0 ($\theta_0(\alpha)$, $\theta_0(\varepsilon)$) in Sec. 2.7.2. The results were obtained with *camera #1.*
Results: The comparison of results using $\theta_0(\alpha) = 0.2125°$ (Eq. (2.10)) and $\theta_0(\varepsilon) = 0.205714°$ (the estimation based on the accuracy of our rotational arm) is presented in Tab. 2.8.

feature	d [cm]	$\theta_0(\varepsilon) = 0.205714°$ $l(\alpha,\beta)$ [cm]	$\theta_0(\varepsilon) = 0.205714°$ $l(\alpha,\beta) - d$ [cm (% of d)]	$\theta_0(\alpha) = 0.2125°$ $l(\alpha,\beta)$ [cm]	$\theta_0(\alpha) = 0.2125°$ $l(\alpha,\beta) - d$ [cm (% of d)]
1	111.5	110.0	-1.5 (-1.3%)	132.2	20.7 (18.6%)
2	95.5	89.6	-5.9 (-6.1%)	102.5	7.0 (7.3%)
3	64.0	59.6	-4.4 (-6.8%)	63.6	-0.4 (-0.6%)
4	83.5	78.8	-4.7 (-5.6%)	87.9	4.4 (5.2%)
5	92.0	89.8	-2.2 (-2.4%)	102.7	10.7 (11.6%)
6	86.5	83.1	-3.4 (-3.9%)	93.6	7.1 (8.2%)
7	153.0	160.2	7.2 (4.7%)	220.7	67.7 (44.2%)
8	130.5	135.5	5.0 (3.8%)	174.3	43.8 (33.6%)
9	88.0	87.6	-0.4 (-0.4%)	99.7	11.7 (13.3%)
10	92.0	90.1	-1.9 (-2.1%)	103.1	11.1 (12.1%)
11	234.5	215.0	-19.5 (-8.3%)	351.3	116.8 (49.8%)
12	198.0	180.4	-17.6 (-8.9%)	263.4	65.4 (33.0%)
13	177.0	188.5	11.5 (6.5%)	281.8	104.8 (59.2%)
		$AVG_{\%}$=4.7% ± 2.7%		$AVG_{\%}$=22.8% ± 19%	

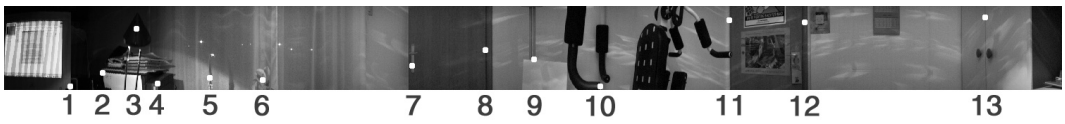


Table 2.8: The comparison of results for two different values of θ_0.

Conclusion: As expected, the results with $\theta_0(\varepsilon) = 0.205714°$ are much better, since it presents the angle for which the robotic arm is rotated in reality. The fact that even a small deviation from real $\theta_0(\varepsilon)$ brings much worse results is also obvious from these results.

2.8.6 Linear versus non-linear model for estimation of angle φ

Experiment background: See the discussion on estimation of the angle φ in Sec. 2.7.2. The results were obtained with *camera #1*.
Results: The comparison of results using $2\varphi = 29.9625°$ (Eq. (2.6)) and $2\varphi = 30.15774565°$ (Eq. (2.8)) is presented in Tab. 2.9.

feature	d [cm]	$2\varphi = 29.9625°$ $l(\alpha,\beta)$ [cm]	$l(\alpha,\beta) - d$ [cm (% of d)]	$2\varphi = 30.15774565°$ $l(\alpha,\beta)$ [cm]	$l(\alpha,\beta) - d$ [cm (% of d)]
1	111.5	110.0	-1.5 (-1.3%)	108.1	-3.4 (-3.0%)
2	95.5	89.6	-5.9 (-6.1%)	88.5	-7.0 (-7.4%)
3	64.0	59.6	-4.4 (-6.8%)	59.2	-4.8 (-7.4%)
4	83.5	78.8	-4.7 (-5.6%)	78.0	-5.5 (-6.6%)
5	92.0	89.8	-2.2 (-2.4%)	88.6	-3.4 (-3.7%)
6	86.5	83.1	-3.4 (-3.9%)	82.2	-4.3 (-5.0%)
7	153.0	160.2	7.2 (4.7%)	155.7	2.7 (1.8%)
8	130.5	135.5	5.0 (3.8%)	132.4	1.9 (1.5%)
9	88.0	87.6	-0.4 (-0.4%)	86.5	-1.5 (-1.7%)
10	92.0	90.1	-1.9 (-2.1%)	88.9	-3.1 (-3.4%)
11	234.5	215.0	-19.5 (-8.3%)	206.7	-27.8 (-11.9%)
12	198.0	180.4	-17.6 (-8.9%)	174.6	-23.4 (-11.8%)
13	177.0	188.5	11.5 (6.5%)	180.4	182.1 (3.4%)
		$AVG_{\%}$=4.7% ± 2.7%		$AVG_{\%}$=5.3% ± 3.5%	

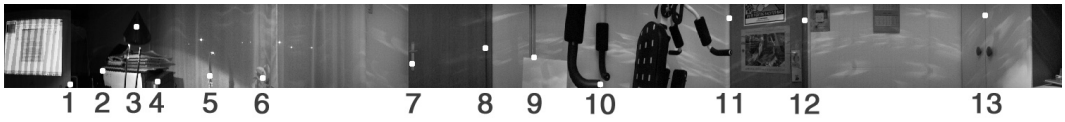


Table 2.9: The comparison of results for two different values of φ: the first one is gained from the linear and the second one from the non-linear model for estimation of angle φ.

Conclusion: The results are not much different, though the results obtained with the linear model are better. Similar results were obtained with *camera #2* in a different room (using again $l(\alpha,\beta)$; see the experiment in Sec. 2.8.8): For $2\varphi = 38.47875°$ (the linear model) the results were $AVG_{\%}$=2.7% ± 2.3%, while for $2\varphi = 38.57170666°$ (the non-linear model) the results were $AVG_{\%}$=3.1% ± 2.6%. Based on these results we can conclude that the linear model is better, at least for a given (estimated) set of parameters.

2.8.7 Repeatability of results — Different room

Experiment background: We want to see if we can achieve similar results as in Sec. 2.8.4, using the same camera (*camera #1*) in a different room?
Results: The results obtained in the different room are presented in Tab. 2.10.

feature	d [cm]	$l(\alpha,\beta)$ [cm]	$l(\alpha,\beta)-d$ [cm (% of d)]
1	63.2	61.5	-1.7 (-2.7%)
2	51.5	50.8	-0.7 (-1.3%)
3	141.0	147.3	6.3 (4.5%)
4	142.0	158.0	16.0 (11.3%)
5	216.0	220.4	4.4 (2.0%)
6	180.0	182.7	2.7 (1.5%)
7	212.0	248.8	36.8 (17.4%)
8	49.0	45.4	-3.6 (-7.4%)
9	49.0	45.4	-3.6 (-7.4%)
10	97.0	95.1	-1.9 (-2.0%)
11	129.5	142.2	12.7 (9.8%)
12	134.0	136.6	2.6 (1.9%)
13	119.0	118.4	-0.6 (-0.5%)
14	156.0	162.5	6.5 (4.2%)
15	91.0	91.2	0.2 (0.2%)
16	97.7	99.3	1.6 (1.6%)
17	111.0	109.3	-1.7 (-1.6%)
18	171.5	175.7	4.2 (2.4%)
19	171.5	182.9	11.4 (6.7%)
		$AVG_{\%}$=4.5% ± 4.5%	

Table 2.10: The results obtained in the different room, but with the same camera as in Sec. 2.8.4.

Conclusion: The overall results are very similar. We can conclude that we can achieve similar accuracy in different rooms. This is also evident from the next experiment, where we have reconstructed the third room with three different cameras. One of them is again *camera #1*. Small differences in results are expected, since each room has its own shape, i.e. the depth distribution around the center of the system is different. And we know how this influences the accuracy, while we are limited with the number of possible depth estimates, which are approximations of the real distances (Sec. 2.7.4).

2.8.8 Repeatability of results — Different cameras

Experiment background: We want to see if we can achieve similar results as in Secs. 2.8.4 and 2.8.7, using different cameras in a different, third room? As mentioned in Sec. 2.7.7, the comparison of results gained using different cameras should not be done at the similar φ, but rather at the similar number of possible depth estimates. This fact is used in this experiment.

Results: The comparison of results for three different cameras is given in Tab. 2.11. Note that for features marked 3, 5, 6, 7, 15, 19, 20 and 21 the real distance d in case of *camera #3* is different from the presented one. The reason for this lies in the vertical view angle of the camera β, which is smaller in comparison to other two cameras. This means that some marked feature points are not seen in the panoramic images generated with *camera #3*, so we have chosen a nearby features with similar distances (see the panoramic image in Tab. 2.12). By all means, in the calculations we have used the correct distances.

feature	d [cm]	*camera #1* $l(\alpha,\beta)$ [cm]	*camera #1* $l(\alpha,\beta)-d$ [cm (% of d)]	*camera #2* $l(\alpha,\beta)$ [cm]	*camera #2* $l(\alpha,\beta)-d$ [cm (% of d)]	*camera #3* $l(\alpha,\beta)$ [cm]	*camera #3* $l(\alpha,\beta)-d$ [cm (% of d)]
1	165.0	162.3	-2.7 (-1.6%)	159.1	-5.9 (-3.6%)	149.0	-16.0 (-9.7%)
2	119.0	118.0	-1.0 (-0.9%)	118.0	-1.0 (-0.8%)	114.1	-4.9 (-4.2%)
3	128.0	133.7	5.7 (4.4%)	130.1	2.1 (1.7%)	119.2	-6.3 (-5.0%)
4	126.5	125.6	-0.9 (-0.7%)	118.3	-8.2 (-6.5%)	114.0	-12.5 (-9.9%)
5	143.0	146.7	3.7 (2.6%)	141.4	-1.6 (-1.1%)	127.8	-13.2 (-9.3%)
6	143.0	151.9	8.9 (6.2%)	141.5	-1.5 (-1.1%)	130.9	-10.6 (-7.5%)
7	142.5	152.7	10.2 (7.2%)	145.0	2.5 (1.7%)	130.9	-11.1 (-7.8%)
8	136.5	141.0	4.5 (3.3%)	135.6	-0.9 (-0.7%)	131.0	-5.5 (-4.0%)
9	104.5	106.8	2.3 (2.2%)	104.6	0.1 (0.1%)	99.1	-5.4 (-5.2%)
10	81.7	79.6	-2.1 (-2.5%)	79.4	-2.3 (-2.8%)	78.6	-3.1 (-3.7%)
11	84.5	80.6	-3.9 (-4.6%)	80.6	-3.9 (-4.6%)	82.3	-2.2 (-2.6%)
12	83.5	82.7	-0.8 (-0.9%)	83.8	0.3 (0.4%)	83.6	0.1 (0.1%)
13	97.0	94.9	-2.1 (-2.2%)	95.7	-1.3 (-1.3%)	93.8	-3.2 (-3.3%)
14	110.0	114.9	4.9 (4.5%)	109.5	-0.5 (-0.5%)	104.9	-5.1 (-4.6%)
15	180.0	191.1	11.1 (6.2%)	165.8	-14.2 (-7.9%)	158.1	-12.9 (-7.5%)
16	124.5	129.9	5.4 (4.3%)	125.2	0.7 (0.6%)	119.2	-5.3 (-4.2%)
17	132.5	132.4	-0.1 (-0.1%)	127.9	-4.6 (-3.5%)	121.8	-10.7 (-8.0%)
18	134.5	136.6	2.1 (1.5%)	131.6	-2.9 (-2.2%)	124.7	-9.8 (-7.3%)
19	113.0	109.4	-3.6 (-3.2%)	107.8	-5.2 (-4.6%)	101.1	-6.9 (-6.4%)
20	125.0	121.6	-3.4 (-2.8%)	118.7	-6.3 (-5.0%)	111.6	-7.4 (-6.3%)
21	130.0	128.8	-1.2 (-1.0%)	121.8	-8.2 (-6.3%)	116.6	-8.4 (-6.8%)
		$AVG_{\%}$=3% ± 2%		$AVG_{\%}$=2.7% ± 2.3%		$AVG_{\%}$=5.9% ± 2.5%	

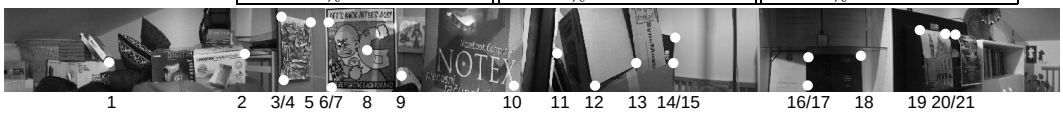


Table 2.11: The results obtained in the third room with three different cameras.

Conclusion: The results show that similar overall accuracy can be achieved if we use different cameras. The reason for somewhat worse results in case of *camera #3* could be attributed to the systematic error presence in the estimation of parameter r, as investigated in the next experiment.

2.8.9 Possibility of systematic error presence in the estimation of r

Experiment background: In Sec. 2.7.2 we have described how the estimation of parameter r is performed. Obviously, it is harder to estimate the location of the optical center in this way, if the view angle is smaller. The problem is even bigger if the camera cannot focus well on the near objects. Let us say that this estimation process is a good starting point for the estimation of system accuracy. We can optimize the estimation of r by minimizing $\mathrm{AVG}_{\%}$: Simply, by letting r go through an interval of possible values around the estimated value, we can calculate $\mathrm{AVG}_{\%}$ for each value of r and, in the end, assign to r the value which minimizes $\mathrm{AVG}_{\%}$. The results were obtained with *camera #3*.
Results: Tab. 2.12 compares the accuracy results before ($r = 356$ mm) and after ($r = 376$ mm) the optimization of parameter r.
Conclusion: We see that the results obtained after the optimization are much better. That r has been underestimated is also obvious from the results of the difference $l(\alpha, \beta) - d$ before the optimization, while they are bigger than normal and they are all, except one, negative.

The same optimization process could of course be used with all other cameras. The remaining error in accuracy could be attributed to:

- the fact that we are limited with the number of possible depth estimates, which are approximations of the real distances (Sec. 2.7.4),
- the error in estimations of other parameters (e.g. α),
- the error due to the lens distortion presence (this matter is addressed in Sec. 2.8.10),
- the human factor (e.g. the distances to the features on the scene are measured manually) and/or
- the possible errors in robotic arm movement.

feature	d [cm]	before optimization $l(\alpha,\beta)$ [cm]	before optimization $l(\alpha,\beta)-d$ [cm (% of d)]	after optimization $l(\alpha,\beta)$ [cm]	after optimization $l(\alpha,\beta)-d$ [cm (% of d)]
1	165.0	149.0	-16.0 (-9.7%)	157.4	-7.6 (-4.6%)
2	119.0	114.1	-4.9 (-4.2%)	120.5	1.5 (1.2%)
3	125.5	119.2	-6.3 (-5.0%)	125.9	0.4 (0.3%)
4	126.5	114.0	-12.5 (-9.9%)	120.4	-6.1 (-4.8%)
5	141.0	127.8	-13.2 (-9.3%)	135.0	-6.0 (-4.2%)
6	141.5	130.9	-10.6 (-7.5%)	138.3	-3.2 (-2.3%)
7	142.0	130.9	-11.1 (-7.8%)	138.2	-3.8 (-2.7%)
8	136.5	131.0	-5.5 (-4.0%)	138.4	1.9 (1.4%)
9	104.5	99.1	-5.4 (-5.2%)	104.6	0.1 (0.1%)
10	81.7	78.6	-3.1 (-3.7%)	83.1	1.4 (1.7%)
11	84.5	82.3	-2.2 (-2.6%)	86.9	2.4 (2.9%)
12	83.5	83.6	0.1 (0.1%)	88.3	4.8 (5.8%)
13	97.0	93.8	-3.2 (-3.3%)	99.1	2.1 (2.1%)
14	110.0	104.9	-5.1 (-4.6%)	110.8	0.8 (0.7%)
15	171.0	158.1	-12.9 (-7.5%)	167.0	-4.0 (-2.4%)
16	124.5	119.2	-5.3 (-4.2%)	125.9	1.4 (1.2%)
17	132.5	121.8	-10.7 (-8.0%)	128.7	-3.8 (-2.9%)
18	134.5	124.7	-9.8 (-7.3%)	131.7	-2.8 (-2.1%)
19	108.0	101.1	-6.9 (-6.4%)	106.8	-1.2 (-1.2%)
20	119.0	111.6	-7.4 (-6.3%)	117.8	-1.2 (-1.0%)
21	125.0	116.6	-8.4 (-6.8%)	123.1	-1.9 (-1.5%)
		$AVG_{\%}$=5.9% ± 2.5%		$AVG_{\%}$=2.2% ± 1.5%	

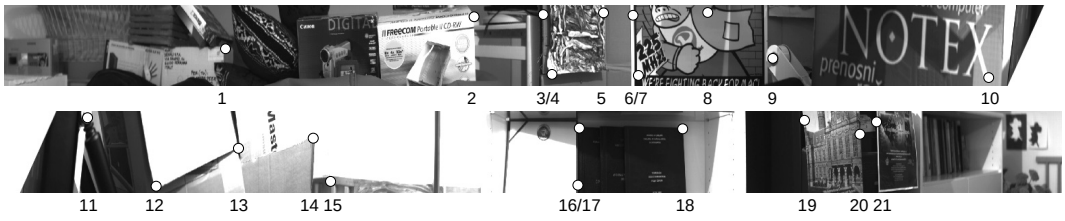


Table 2.12: The comparison of results before and after the optimization of parameter r.

2.8.10 Influence of lens distortion presence on the reconstruction accuracy

Experiment background: Lens distortion is a well known property of camera lens, which causes images to be spherised at their center. This basically means that the pixels that should be on the image edge are actually moved more towards the center of the image. How much they are moved towards the center depends on the camera field of view. Bigger is the field of view, bigger is the error due to the lens distortion. For *camera #3* the maximal error due to the lens distortion is small, only 0.8 pixel in 160×120 pixel images. On the other hand, for *camera #2* the maximal error due to the lens distortion is already 5 pixels in 160×120 pixel images. Fig. 2.19 nicely illustrates this fact. (The error in 640×480 pixel images is 4 times bigger.)

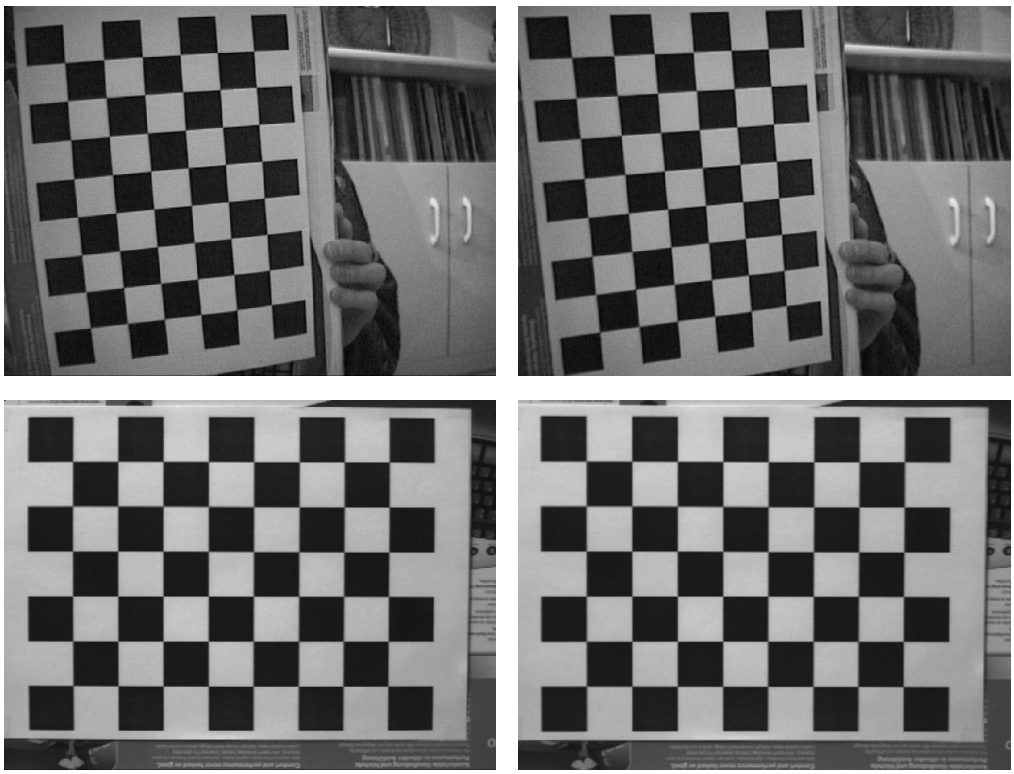


Figure 2.19: Each row of images shows distorted image (left) and undistorted image (right), after the distortion has been suppressed. The images in the top row have been taken with *camera #2*, while the images in the bottom row have been taken with *camera #3*. The resolution of all presented images is 640×480 pixels.

Since the best results obtained with *camera #3* are already very good (Sec. 2.8.9) and the size of distortion here is very small, we use the camera with the widest view angle, i.e. *camera #2*, in this experiment. Fig. 2.20 shows the camera

model gained after the calibration process over a set of 640×480 pixel images [54]. We have used this model to undistort the captured images before they were merged into the panoramic images.

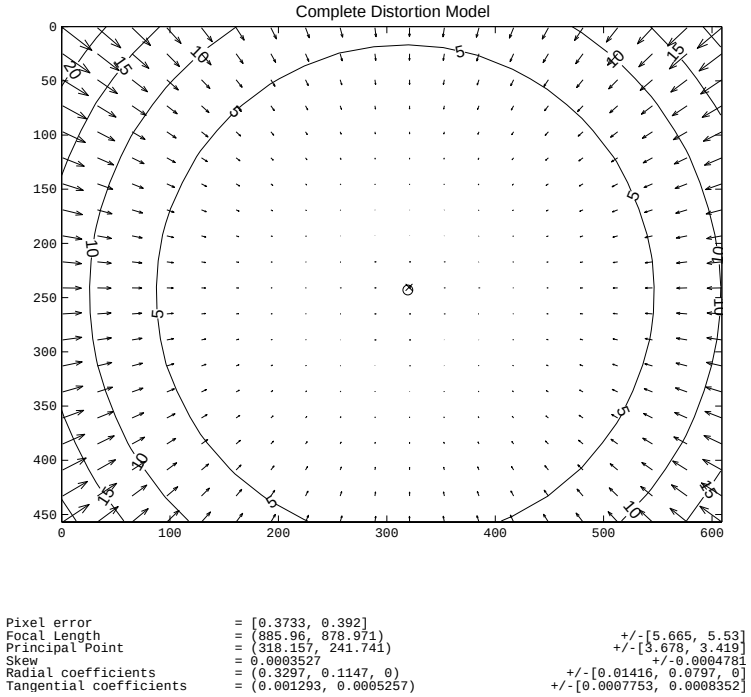


Figure 2.20: The camera model gained after the calibration process over a set of 640×480 pixel images [54]. Note that the values of estimated parameters (that are given in pixels) are 4× smaller for 160×120 pixel images, which we use for generating panoramic images, and that the errors given in the right bottom side of the figure are approximately three times the standard deviations (for reference). Having that in mind, we see that the principal point in 160×120 pixel images is right in the middle of the images. The values on the curves in the figure present the errors in pixels due to the lens distortion.

As we mentioned, the pixels that should be on the image edge are actually moved more towards the center of the image, which means that after the distortion is corrected the camera field of view gets smaller (Fig. 2.19). The new vertical field of view α_{new} was estimated using a simple observation about the part of the scene (number of pixels n) that disappeared from the image due to the distortion correction (similar to Eq. (2.6)):

$$\alpha_{new} = \frac{W - n}{W} \cdot \alpha; \text{ for } n = 9 \Rightarrow \alpha_{new} = 37.48575^{\circ}.$$

W is again the width of the captured image.

We also know that different α brings different r (Sec. 2.7.2), so we have to correct the size of parameter r as well:

$$r_{new} = \frac{\alpha_{new}}{\alpha} \cdot r = 293 \text{ mm}.$$

Similarly, all other parameters could be estimated if they are needed, e.g. the focal length f could be estimated from Eq. (2.2). But θ_0 stays the same as it still (even after the distortion correction) represents the angle for which the rotational arm has been moved between each two successively captured images.

Results: By using undistorted images to generate panoramic images and the new values of parameters, we obtain results presented on the right side in Tab. 2.13. For comparison, on the left side the results using distorted sequence are presented. In case of undistorted sequence, we have used $2\varphi = 37.04933695°$ as it ensures a similar number of possible depth estimates as the basic settings of *camera #2* (Sec. 2.7.7). Note that for the feature marked 11 the real distance d in case of undistorted sequence is different from the presented one. The reason for this lies in the fact that the normalized correlation procedure has been unable to find the appropriate corresponding point, so we have chosen a nearby feature with a similar distance. By all means, in the calculations we have used the correct distance.

Conclusion: We can conclude that processing undistorted images brings better results, though quite comparable. Having in mind that undistorting the sequence means that more processing time is needed (for instance, in Matlab (running on a 2.0 GHz Intel PIV PC) it takes a few hours to process 1501 images of size 160×120 pixels), perhaps we should be satisfied with the results gained using the distorted sequence. Another drawback of undistorted images is that they are more blurred in comparison to distorted originals (Fig. 2.19). Nevertheless, by using cameras with even wider field of view the distortion gets more obvious, and consequently we cannot always neglect its presence.

feature	d [cm]	distorted sequence $l(\alpha,\beta)$ [cm]	distorted sequence $l(\alpha,\beta)-d$ [cm (% of d)]	undistorted sequence $l(\alpha,\beta)$ [cm]	undistorted sequence $l(\alpha,\beta)-d$ [cm (% of d)]
1	165.0	159.1	-5.9 (-3.6%)	165.6	0.6 (0.4%)
2	119.0	118.0	-1.0 (-0.8%)	118.4	-0.6 (-0.5%)
3	128.0	130.1	2.1 (1.7%)	123.7	-4.3 (-3.4%)
4	126.5	118.3	-8.2 (-6.5%)	122.3	-4.2 (-3.3%)
5	143.0	141.4	-1.6 (-1.1%)	139.5	-3.5 (-2.5%)
6	143.0	141.5	-1.5 (-1.1%)	139.5	-3.5 (-2.4%)
7	142.5	145.0	2.5 (1.7%)	143.8	1.3 (0.9%)
8	136.5	135.6	-0.9 (-0.7%)	133.7	-2.8 (-2.0%)
9	104.5	104.6	0.1 (0.1%)	103.8	-0.7 (-0.7%)
10	81.7	79.4	-2.3 (-2.8%)	77.1	-4.6 (-5.6%)
11	84.5	80.6	-3.9 (-4.6%)	78.4	-5.1 (-6.1%)
12	83.5	83.8	0.3 (0.4%)	81.7	-1.8 (-2.1%)
13	97.0	95.7	-1.3 (-1.3%)	96.5	-0.5 (-0.6%)
14	110.0	109.5	-0.5 (-0.5%)	106.3	-3.7 (-3.4%)
15	180.0	165.8	-14.2 (-7.9%)	173.0	-7.0 (-3.9%)
16	124.5	125.2	0.7 (0.6%)	122.7	-1.8 (-1.4%)
17	132.5	127.9	-4.6 (-3.5%)	129.4	-3.1 (-2.3%)
18	134.5	131.6	-2.9 (-2.2%)	133.6	-0.9 (-0.7%)
19	113.0	107.8	-5.2 (-4.6%)	109.8	-3.2 (-2.8%)
20	125.0	118.7	-6.3 (-5.0%)	122.4	-2.6 (-2.0%)
21	130.0	121.8	-8.2 (-6.3%)	126.1	-3.9 (-3.0%)
		$AVG_{\%}$=2.7% ± 2.3%		$AVG_{\%}$=2.4% ± 1.6%	

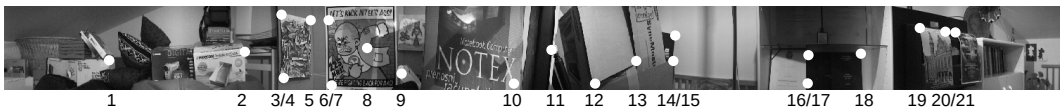


Table 2.13: The comparison of results obtained without and with the lens distortion correction.

2.9 Summary

We have presented a comprehensive analysis of our mosaic-based system for construction of depth panoramic images using only one standard camera.

The conclusions about the system effectiveness and its accuracy are well exposed in Secs. 2.7 and 2.8. In the end, let us write only the main conclusion, which answers to the question written at the very beginning of this chapter (Sec. 2.1.1): Can the system be used for robot localization and navigation in a room? According to the accuracy achieved the answer is: Yes!

The main conclusions made in this chapter are summarized in Sec. 4.2.

Chapter 3

Real Time Extension

3.1 Introduction

3.1.1 Motivation

Real time panoramic depth imaging from multiperspective panoramas is an issue that is not well covered in the literature. There have been attempts or discussions [1] about it, but nothing has been done in practice so far, at least not by using the mosaicing concept, i.e. the multiperspective panoramas.

There are of course other possibilities how to acquire panoramic depth images, as already mentioned in Sec. 2.2. Let us briefly present the system called SOS (Stereo Omnidirectional System) [30, 41, 42], which we intentionally left out from discussion in Sec. 2.2, since it is much more related to the topic of the second part of the thesis. The SOS system uses standard cameras and produces panoramic depth images in real time, but here all the similarities with our system end. The system consists of 20 stereo units and each unit consists of 3 standard cameras. The units are arranged on each plane of a regular icosahedron. The basic principle for generating panoramic depth images is as follows: Each unit captures three images from which a normal (not panoramic) depth image is computed on a computer with more CPU units. Then the generated depth images are registered into the panoramic depth image. Obviously, the authors use a different concept than we do: they first build standard depth images and then the panoramic depth image, while we first build panoramic images and then the panoramic depth image. So, from this point, we are interested only in panoramic depth imaging from multiperspective panoramas.

In order to capture stereo panoramic images of dynamic scenes Peleg et al. [39] presented a theory for construction of a special mirror and lens such that viewing the scene through this mirror or lens creates the same rays as those used with the rotating cameras. To our knowledge the systems are still not constructed, while, according to authors, a lot of practical issues need to be solved before a camera is built.

In the first part of the thesis we have presented a system for capturing panoramic depth images with a single standard camera. A stereo pair of images is captured while the camera rotates around the center of the system in a horizontal plane. The motion parallax effect which enables the reconstruction can be captured because of the offset of the cameras' optical center from the systems' rotational center. The camera is moving around the rotational center in angular steps corresponding to one vertical pixel column of the captured standard image. A symmetric pair of panoramic stereo images are generated so that one column on the right side of the captured image contributes to the left eye panoramic image and the symmetric column on the left side of the captured image contributes to the right eye panoramic image. This system however cannot generate panoramic stereo pair in real time. To illustrate this fact, we can write down the following example from practice: if the system builds a panoramic stereo pair from standard images with resolution of

160×120 pixels, using a camera with the horizontal view angle $\alpha = 34°$, it needs around 15 minutes to complete the task (Sec. 2.7.1).

Generally, mosaic-based procedures for building panoramic images can be marked as non-central, they do not execute in real time and they give high resolution results (Sec. 2.2). Thus mosaicing is not appropriate for capturing dynamic scenes. The main advantage of these procedures over other panoramic imaging systems (like catadioptric systems [52]) is the ability to generate high resolution results. But high resolution results are essential for effective depth recovery based on the stereo effect.

3.1.2 Structure of the chapter

In the next section we discuss how the time needed for the generation of a symmetric panoramic stereo pair can be dramatically reduced. In Sec. 3.3 we go even further and explain how we can achieve real time execution. Sec. 3.4 presents the depth reconstruction equation for the new setup. The epipolar constraint is discussed in Sec. 3.5. The evaluation of results is given in Sec. 3.6. We end this chapter with summary in Sec. 3.7.

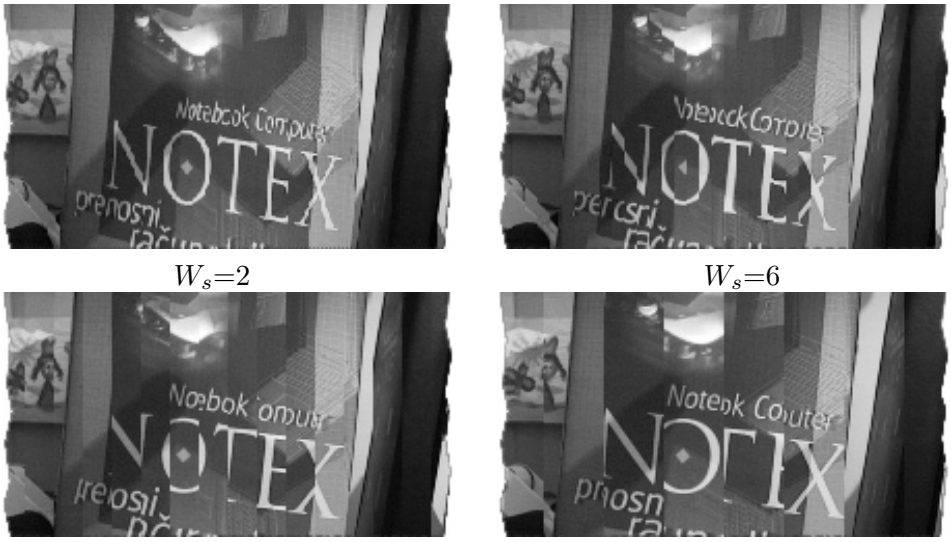
3.2 Building panoramic images from wider stripes

Let us begin this section by pointing to the description of the old sensor geometry, which is given in Sec. 2.4.

The first idea about how to capture a stereo pair quicker is to generate panoramic images from wider vertical stripes instead of just one column.

This task is by all means much faster, but at the same time we have to make a compromise between the speed of the capturing task and the quality of the stereo pair (Fig. 3.1). First of all, the wider the stripes are, the more obvious are the stitches between the stripes in the panoramic image. Then, with the wider stripe the difference between the amount of lens distortion of the first and the last pixel columns in the stripe is more noticeable. But the real problem arises from the fact that stripes introduce a property, which influences the coverage of the scene, as presented in the next section. As we show in the experimental results (Sec. 3.6), we are satisfied with the result when we use 14 columns wide stripes and we think that it represents a good compromise. This statement is naturally highly related to the camera that we use.

In case the horizontal view angle of the camera is 34°, where 14 columns represent the angle of 2.879999°, the building process takes approximately 14 times less, i.e. around a minute (Sec. 2.7.1).



W_s=2 W_s=6

W_s=14 W_s=29

Figure 3.1: Panoramic images gained with the camera with the horizontal view angle α=34° using different stripe widths. W_s gives the number of pixel columns within the stripe (the width of the stripe). Consequently, bigger W_s means bigger shift angle between two successive images that contribute to panoramas, i.e. faster execution of the building process and worse quality of the generated panoramas.

3.2.1 Property of using stripes

If we observe the panoramic image built from stripes closely, we can notice that the image is not perfect. In this case we are not referring to stitches nor lens distortion. These problems are present, but are not too disturbing. Another problem can be noticed on close objects on the scene, which have a nice texture on them (like text). In such a case we can see that some points on the scene are not captured (Fig. 3.2). Of course, this is partly because of the fact that we are dealing with the images, which are discrete (for example: we cannot take a half of a pixel), but if we take a look at the geometry of the system (Fig. 2.2), we can see that this is not the only reason. If we consider two successive steps of the system, we can see that the stripes that contribute to the panoramic image do not cover all the scene.

The drawings in Figs. 3.3, 3.5, 3.6, 3.7 and 3.8 show the formation of the panoramic image with respect to the light rays (compare with Fig. 2.2). All mentioned drawings are done for the actual case of horizontal view angle $\alpha = 34°$ and are **drawn in the same ratio** to illustrate the problem of uncaptured scene points. The black circle represents the camera path, the light circle represents the viewing circle, the black dot in the middle of the circles represents the center of rotation, an intersection of black line from the center of rotation to the black circle and the

Figure 3.2: The wider is the stripe, the more scene points are not captured in the panoramic image: the left panoramic image was built from single columns, while the right panoramic image was built from 14 columns wide stripes. Note how very distant scene points are well captured in both examples and how some nearby scene points (text on the box) are not captured in the second example.

black circle represents the position of the optical center on the camera path, a black line from the optical center going outwards represents the light ray that is in the middle of light rays that form one pixel column or stripe in each captured image that contribute to the panoramic image, the lighter line tangent to the viewing circle is the extension of that light ray towards the virtual camera and a bit darker lines from the optical center going outwards represent bordering light rays that form one pixel column or stripe in each captured image that contribute to the panoramic image.

Let us first examine the example given on the right side in Fig. 3.2, where the shift angle corresponds to the stripe of 14 pixel columns of the captured image (W_s=14): $\alpha = 34°$, $r = 30$ cm, $2\varphi_{\min} = 27.082501°$, $2\varphi_{\max} = 32.842499°$ (these two values $\varphi_{\min}$ and $\varphi_{\max}$ define the stripe, which consists of more pixel columns, similarly as φ defined one pixel column; for more information see Sec. 3.4), $\theta_0 = W_s \cdot 0.205714°$. The drawing is presented in Fig. 3.3. Since each stripe of the captured image is formed from more light rays, which also circumscribe a similar angle to the shift angle (see the discussion on estimation of the angle θ_0 ($\theta_0(\alpha)$, $\theta_0(\varepsilon)$) in Sec. 2.7.2 and the experiment in Sec. 2.8.5), the middle light ray (black line) and the border light rays (gray lines) are presented for each stripe that contributes to the panoramic image. We can observe two properties on this drawing (a detail of the drawing in the rectangular area is given in Fig. 3.4): 1) We really cannot capture all the scene points, since there is a gap between two successive stripes that contribute to the panoramic image. 2) With growing distance from the rotational center the bordering light rays of two successive stripes are getting nearer until they intersect (Fig. 3.4). Generally, the shift angle and the angle corresponding to the stripe of the captured image should be the same, but they are not due to the limited accuracy of our rotational arm. In this general, ideal case, the bordering light rays of two successive stripes would be parallel (for example, imagine that both angles are 90°). Since we deal here with two independent discretizations, namely with the discretization of the standard image (pixels; $\theta_0(\alpha)$ given in Sec. 2.7.2) on one side and with the discretization of the rotational arm (the minimal step of the rotational arm defined by its maximal accuracy; $\theta_0(\varepsilon)$ given in Sec. 2.7.2) on the other side,

it is almost impossible to achieve the ideal case ($\theta_0(\alpha) = \theta_0(\varepsilon)$). Because of these discretizations the bordering light rays of two successive stripes are normally getting nearer ($\theta_0(\alpha) > \theta_0(\varepsilon)$) or further appart ($\theta_0(\alpha) < \theta_0(\varepsilon)$).

Already only the first property reveals that there is always a part of the scene which cannot be captured. And since the effect loses on significance with a bigger distance from the rotational center, the distant points on the scene look well captured (Fig. 3.2), also since the bigger distance to the object results in lower resolution of the object. On the other hand, in our case, the bordering light rays of two successive stripes are getting nearer with bigger distance, which means that this effect is getting even smaller with increased distance. This also means that in the far distance the bordering light rays are intersecting, but since the intersection point is far away the effect of repeatability of scene points in the panoramic image is not noticeable, again because with the bigger distance of the system from the object its resolution in the image gets lower.

This means that a similar effect should be present when we build panoramic images from only one pixel column of each captured image, while one pixel width is not infinitesimal, although the part of the panoramic image presented on the left side in Fig. 3.2 looks faultless. In case presented in Fig. 3.5 the shift angle corresponds to one pixel column of the captured image (W_s=1): $\alpha = 34°$, $r = 30$ cm, $2\varphi = 29.9625°$, $\theta_0 = 0.205714°$. The description of the building process is the same, the main difference is that the stripe is now only one pixel column wide. This means that now the gaps are much smaller (compare Figs. 3.3 and 3.5).

So, we can write one more conclusion with respect to this property. Namely, the wider is the stripe, the more scene points are not captured in the panoramic image (Figs. 3.2, 3.3 and 3.5). And this holds regardless of the position of the stripe in the captured image. We can take a stripe from the middle of the captured image (Fig. 3.6) or from the edge of it, but the presented property would still be seen on the resulting panoramic image. Naturally, by using single columns we achieve best possible result (Fig. 3.2), though still not perfect, since the described property still holds, but is not so obvious (Fig. 3.5). By using smaller r the gaps are also smaller, which means that with smaller r we cover more scene points (compare Figs. 3.3 and 3.7). Other properties of r are already discussed in Sec. 2.7.2.

From Fig. 3.8 it is evident that if we build panoramic image from images captured in the center of rotation, i.e. without the motion parallax effect, some points on the scene are captured twice. Thus, the bordering light rays in this case are intersecting already in the center of rotation. But since the overlapping is very small this is not noticeable in the generated results. On the other hand, in practice, some scene points could also be missing. But really small deviations (generally not noticeable by the human eye) of the generated panoramic images from the scene should be attributed to the fact that the images are discrete (for example: we cannot take a half of a pixel). Much more noticeable are the stitches and the lens distortion. According to Fig. 3.8 we can conclude that if we want to capture the parallax effect,

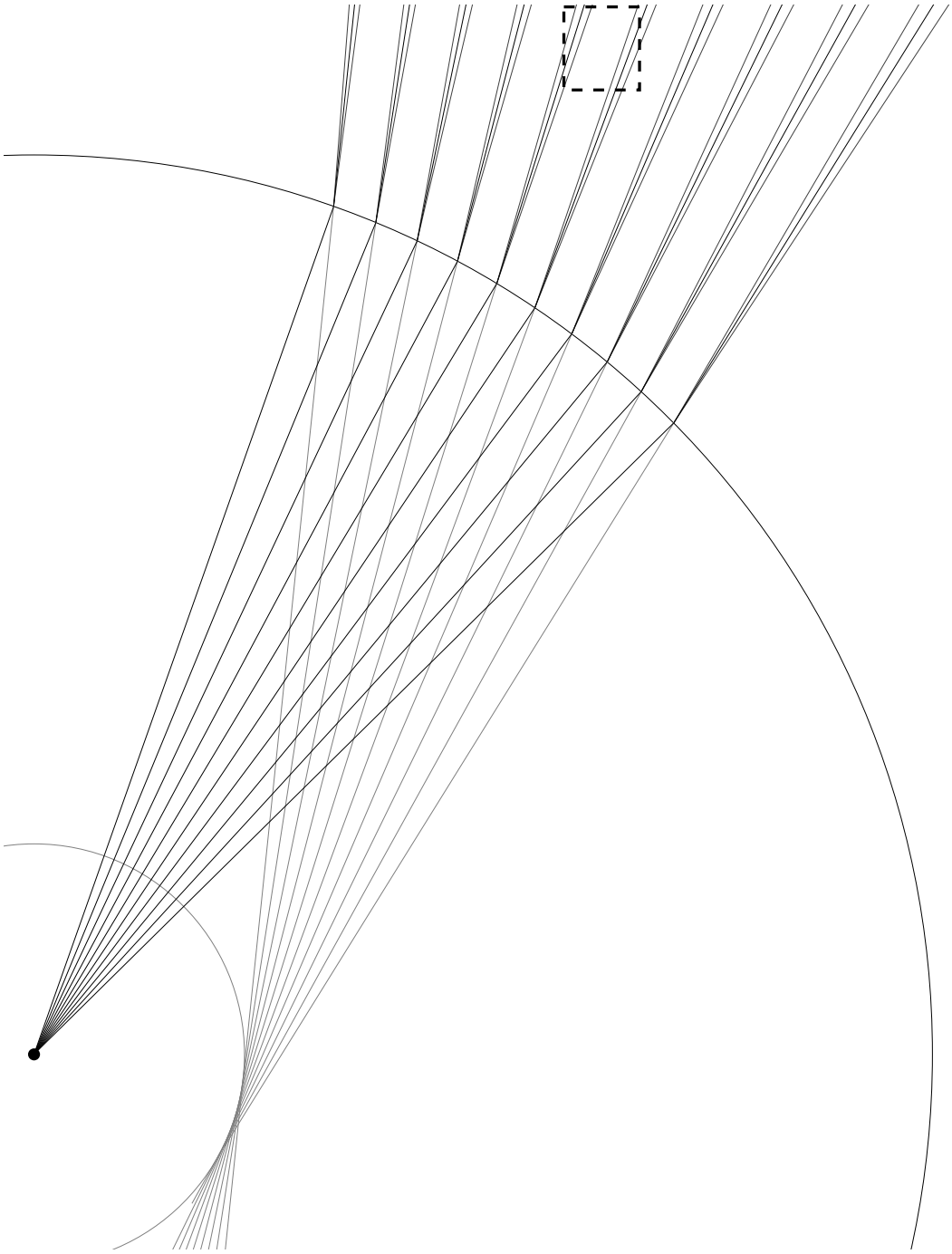


Figure 3.3: The drawing shows the formation of the panoramic image with respect to the light rays. In this case the shift angle corresponds to 14 pixel columns of the captured image. For detailed description see Sec. 3.2.1. The detail in the rectangular area at the top is given in Fig. 3.4.

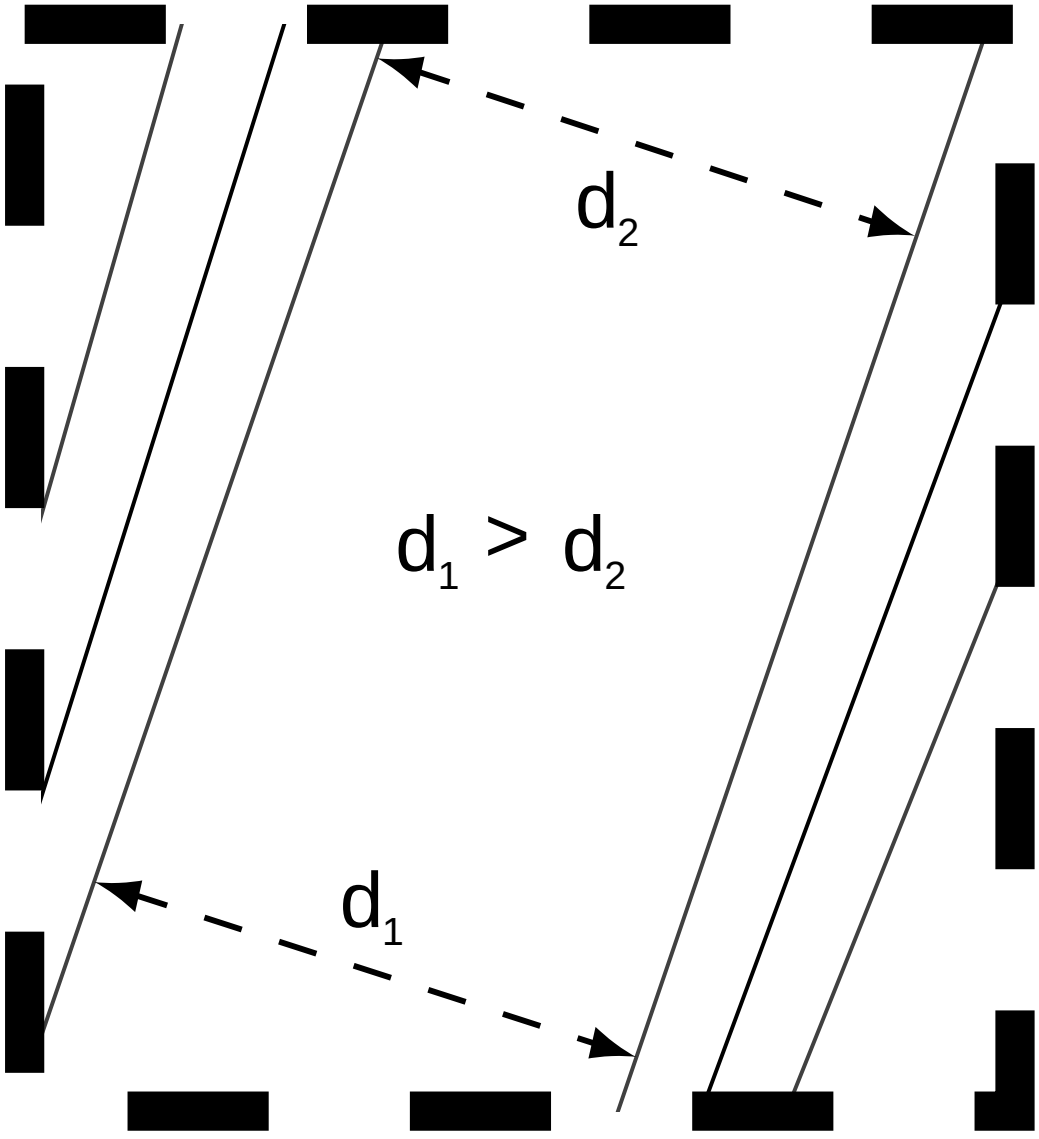


Figure 3.4: The detail of the drawing presented in Fig. 3.3. For detailed description see Sec. 3.2.1..

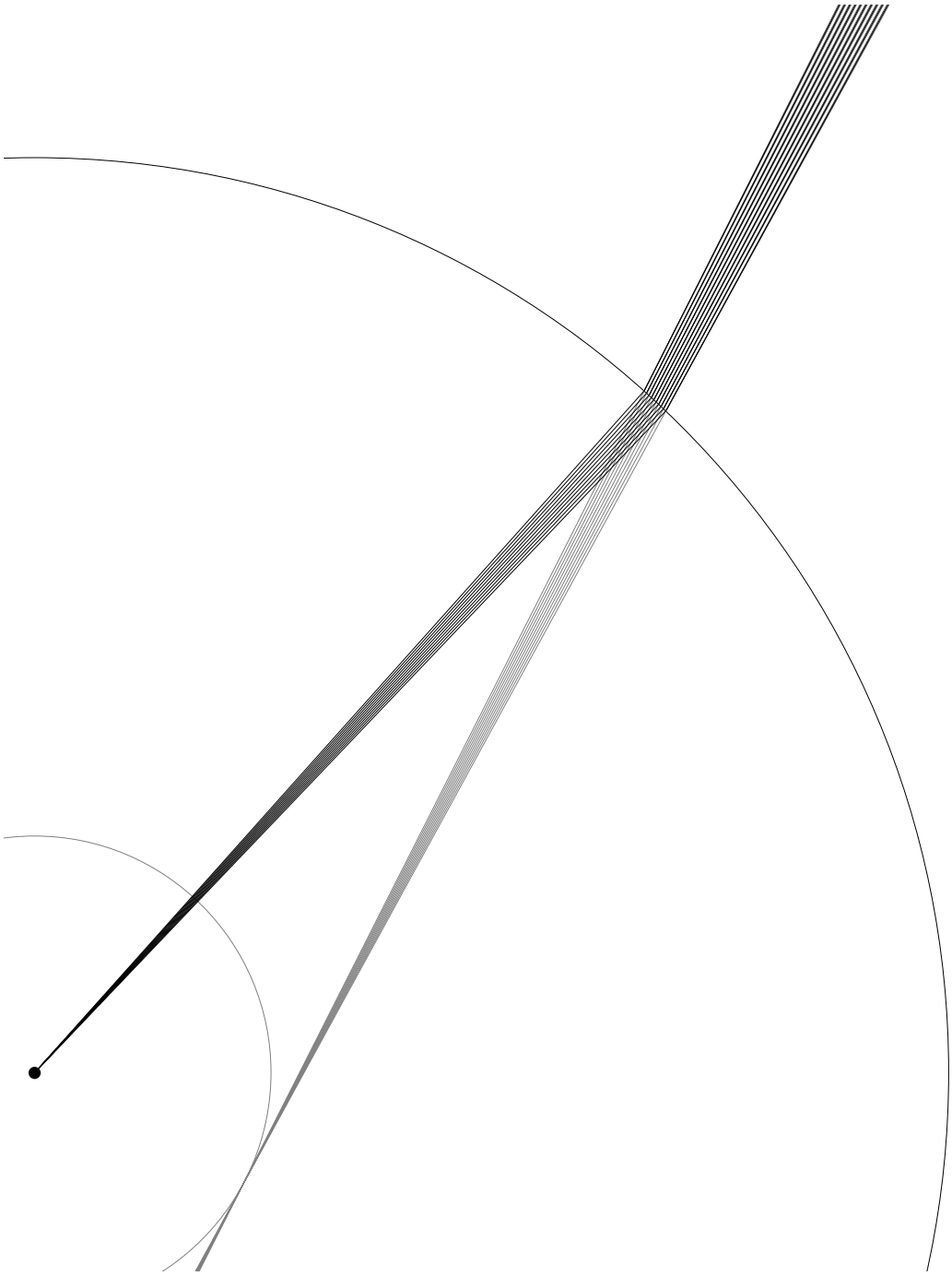


Figure 3.5: The drawing shows the formation of the panoramic image with respect to the light rays. In this case the shift angle corresponds to one pixel column of the captured image. For detailed description see Sec. 3.2.1.

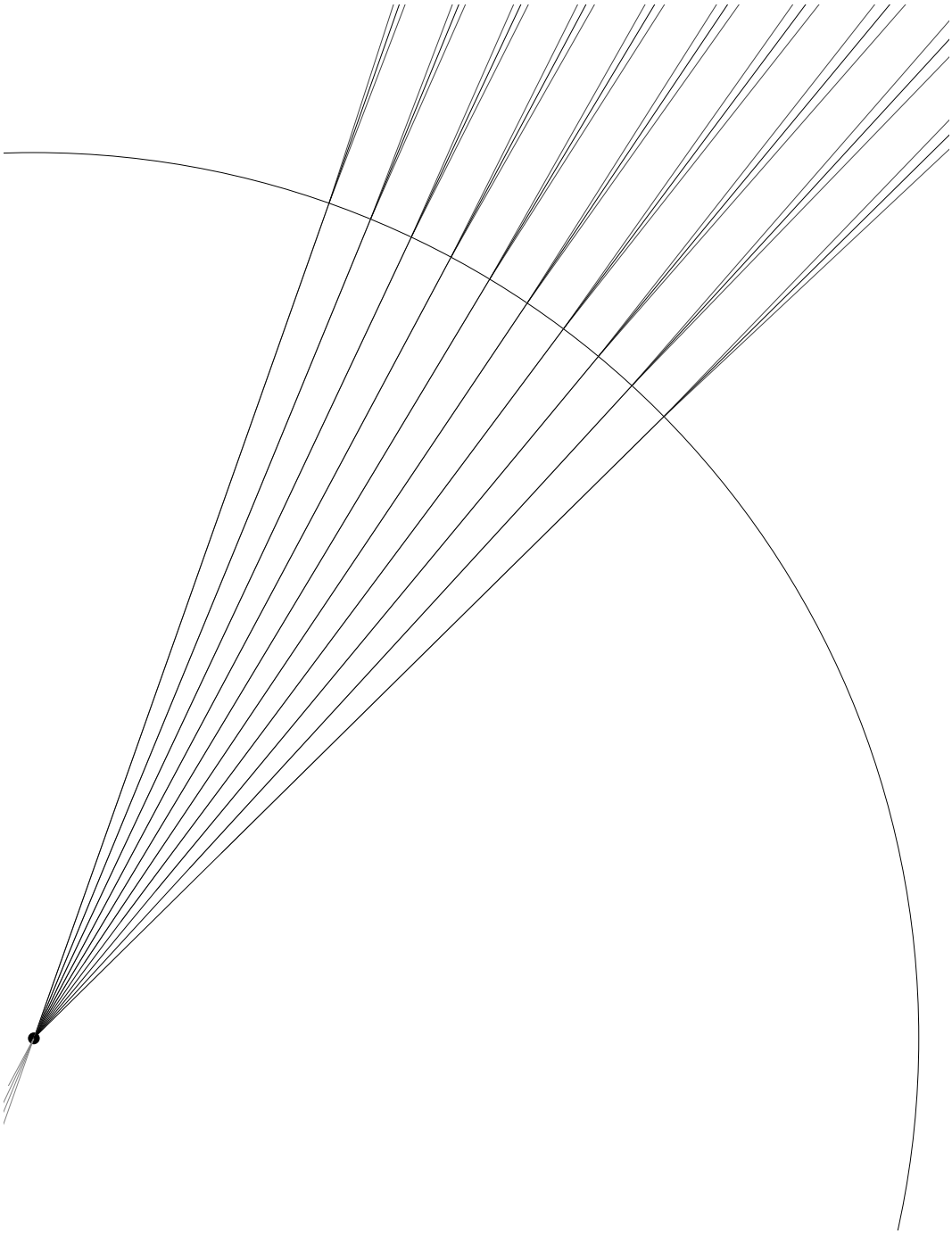


Figure 3.6: The drawing shows the formation of the panoramic image with respect to the light rays. In this case the shift angle corresponds to W_s=14 pixel columns taken from the center of the captured image: $\alpha = 34°$, $r = 30$ cm, $2\varphi_{\min} = -2.879999°$, $2\varphi_{\max} = 2.879999°$, $\theta_0 = W_s \cdot 0.205714°$. For detailed description see Sec. 3.2.1.

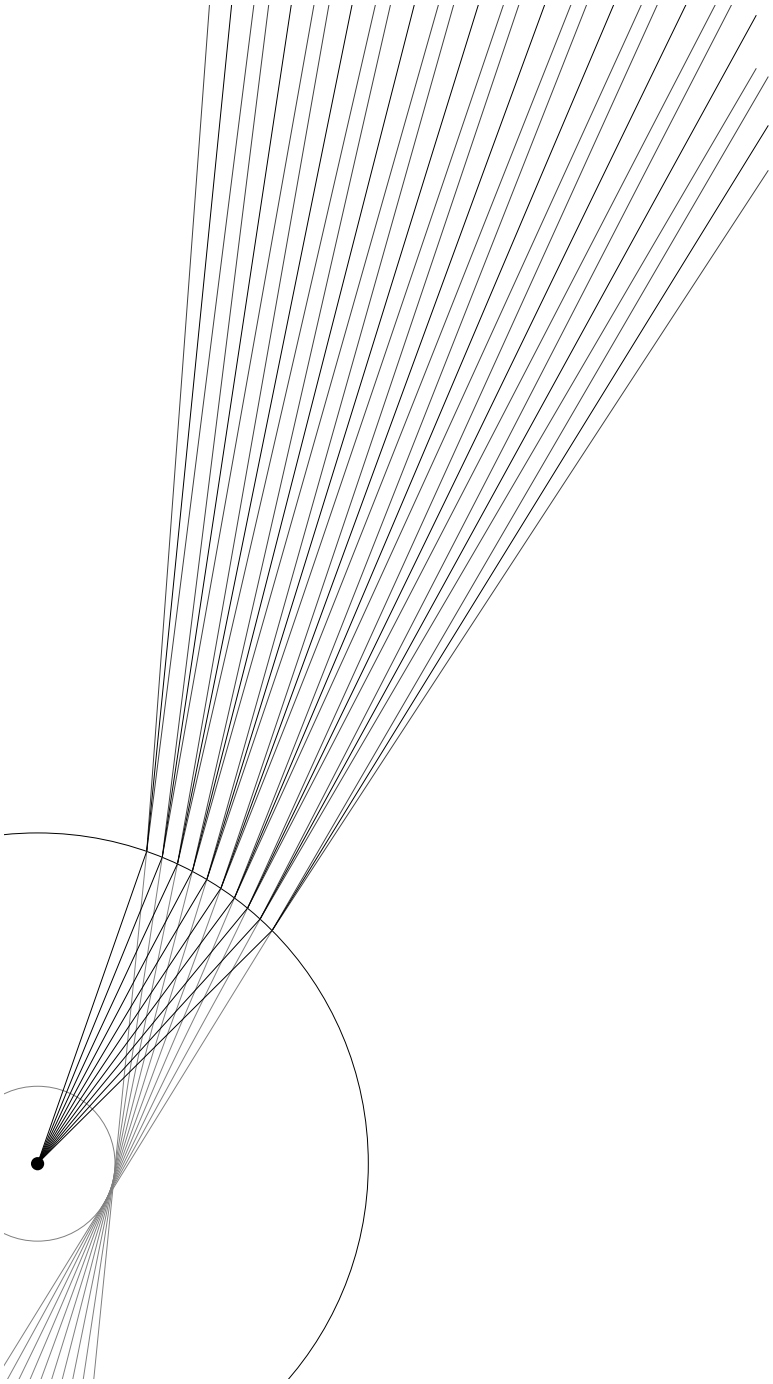


Figure 3.7: The drawing shows the formation of the panoramic image with respect to the light rays. In this case the shift angle corresponds to 14 pixel columns of the captured image, but r=10 cm (3 times smaller than in Fig. 3.3). For detailed description see Sec. 3.2.1.

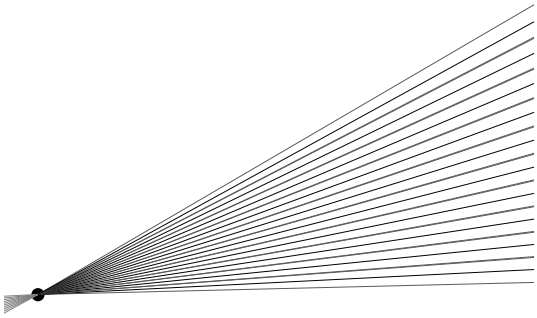


Figure 3.8: The drawing shows the formation of the panoramic image with respect to the light rays. The images are captured in the center of rotation, i.e. without the motion parallax effect. In this case the shift angle corresponds to W_s=14 pixel columns taken from the center of the captured image: $\alpha = 34°$, $r = 0$ cm, $2\varphi_{\min} = -2.879999°$, $2\varphi_{\max} = 2.879999°$, $\theta_0 = W_s \cdot 0.205714°$. For detailed description see Sec. 3.2.1.

we have to accept the fact that not all the scene points are captured.

3.3 Achieving real time

If we use stripes instead of single columns from each captured image, we can drastically reduce the time needed to build a panoramic image. But we still cannot build a panoramic image in real time.

The idea for a real time panoramic sensor is actually very simple. In our old system (Chapter 2) the panoramic image is build by means of moving the standard camera for a very small angle along a predefined circular path. If we could have a camera on each position on the circular path, we could build the panoramic image in real time. But unfortunately in practice we cannot put so many cameras so close together (with respect to a reasonable size of radius r). If we build a panoramic image from captured images with the resolution of 160×120 pixels, then we have to put the cameras with the horizontal view angle $\alpha = 34^\circ$ 0.205714° apart from each other and we need $360/0.205714 \doteq 1750$ cameras.

In the case when we use stripes, the presented numbers get more reasonable. A 14 column stripe suggests that the cameras would be 2.879999° apart from each other and we would need 125 cameras to cover the whole circular path. If we use a camera with a wider horizontal view angle (e.g. $\alpha = 90^\circ$), we need less cameras (e.g. 46). The new sensor does not need any moving parts, which means that we do not deal with mechanical vibrations nor are we limited with the radius of the circle on which the cameras are fixed. The last statement about the radius enables us to make the sensor out of standard cameras that are available on the market.

Fig. 3.9 shows the drawing of a real time sensor. The sensor is made out of cameras presented in Fig. 2.1 ($\alpha = 34^\circ$), where the angle between successive cameras corresponds to 14 columns of the captured image with the resolution of 160×120 pixels. This camera is probably much to big to be used in the real sensor (the ground plan dimensions of the camera are 14.5×4.5 cm), since r in this case should be set to at least 100 cm. On the other hand, we could set r to 30 cm and this implies that the camera should be smaller for $3.\overline{3}$ times (in this case the ground plan dimensions of the camera should be 4.35×1.35 cm). We have simulated such sensor with our camera (Fig. 2.1) in order to determine its accuracy. The experimental results are given in Sec. 3.6, where we have also tested other cameras, different widths of the stripes etc. If we built the sensor, we would normally use board cameras, where the lens, as the biggest part, is even smaller and the board to which the lens is attached is flexible, so that we are able to put the lenses completely together. Even if such a camera is not available on the market, it can certainly be produced or we have to make a compromise between the physical size of the camera and the size of r. For this purpose we have made a function, which gets camera parameters, together with its dimensions, as an input and produces drawings like the one in Fig. 3.9. Its

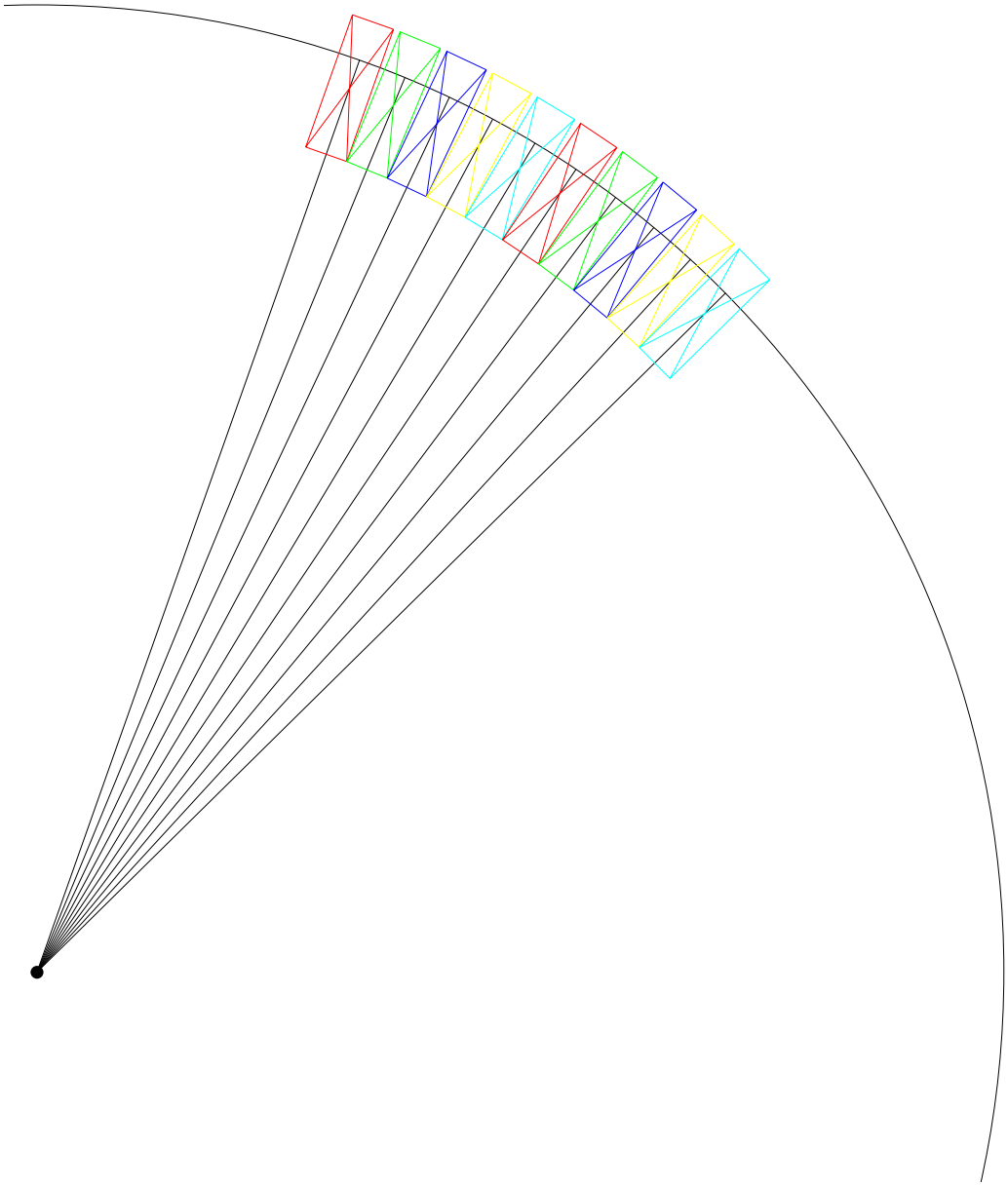


Figure 3.9: The drawing of a real time sensor. The boxes with crosses represent the physical cameras positioned with respect to the center of rotation. Optical center of each camera lies in the intersection between the line going outwards from the rotational center and the circle with radius r. The sensor is made out of cameras presented in Fig. 2.1 ($\alpha = 34°$) and is presented in 1:1 ratio between parts of the sensor. For detailed description see Sec. 3.3.

goal is to determine the size of the minimal r for the given camera. We have not investigated what kind of cameras are available on the market, but nevertheless we determined the size of r for the Neocom board camera ASH-23B/AH-23B equip with SONY CCD sensor (*http://www.ncctv.com/bw_board_camera.htm*) with the ground plan dimensions of the camera being 2.3×2.7 cm. We assumed that the field of view is $\alpha = 34^\circ$ (the same as in the camera we used), since a number of different lenses are possible. r was determined to be at least 50 cm. At Neocom they also claim that they have round type board cameras for the use in dental clinics. This suggest that smaller cameras are available.

It is not our intention in the thesis to evaluate the sensor from the financial point of view, but we can make a rough approximation with respect to the number of cameras that are needed to build the sensor. Though we have not gained the information about the price of the board camera ASH-23B/AH-23B, we can estimate the average price of the board camera to be around 100 USD. This means that 125 cameras would cost 12.500 USD.

3.4 Stereo reconstruction from stripes

By following the sinus law for triangles, we can simply write the equation for the depth l as (Fig. 2.2):

$$l = \frac{r \cdot \sin\varphi}{\sin(\varphi - \theta)}. \tag{3.1}$$

This equation holds if we do the reconstruction based on the symmetric pair of stereo panoramic images built from one pixel columns of the captured image.

But when we use stripes, we have to adopt the equations according to the new building process. In this case we take symmetric stripes instead of symmetric columns from the captured image. While the column was defined by the angle φ, the stripe is defined by two such angles: $\varphi_{\min}$ and $\varphi_{\max}$. On the left eye panoramic image we can assign the angle φ_l to each pixel within the stripe: $\varphi_{\min} \leq \varphi_l \leq \varphi_{\max}$. After finding the corresponding point on the right eye panoramic image, we can evaluate the angle φ_r in the same manner, according to the position of the corresponding pixel within the stripe: $\varphi_{\min} \leq \varphi_r \leq \varphi_{\max}$. Now let us assume that we can still calculate the angle θ as in Sec. 2.6 (see the next section to clear the issue of why we can assume this):

$$2\theta = dx \cdot \theta_0, \tag{3.2}$$

where dx is the absolute value of the difference between x coordinates of the corresponding points in the left eye panoramic image and in the right eye panoramic image, while θ_0 is the angle corresponding to one pixel column of the captured image and consequently the angle for which we have to move the robotic arm if we build

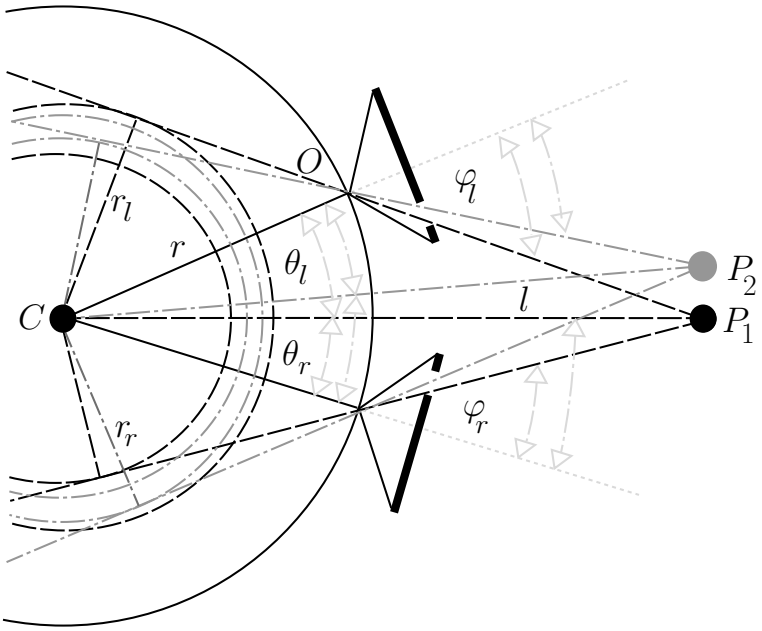


Figure 3.10: Angles θ_l and θ_r are related to angles φ_l and φ_r as presented in Eq. (3.4). Here the relationship is illustrated for two scene points.

the panoramic images from only one column of the captured image. Using analogy for this equation and having in mind that we are building the panoramic images from stripes, we can write the following equation (Figs. 2.2 and 3.10):

$$2\theta = \theta_l + \theta_r. \tag{3.3}$$

When we use one column instead of stripes then $\theta_l = \theta_r$ (Fig. 2.2), but this is not necessary true if we use stripes. In general these two values are different, but the property following from the equation

$$\frac{\theta_l}{\theta_r} = \frac{\varphi_l}{\varphi_r} \tag{3.4}$$

shows that the ratio of these two values is related to the angles φ (Fig. 3.10). The bigger φ_l gets, the bigger gets the corresponding θ_l. Now we can simply express θ_r and θ_l from Eqs. (3.2), (3.3) and (3.4) as:

$$\theta_r = \frac{dx \cdot \theta_0}{(1+\frac{\varphi_l}{\varphi_r})}$$
$$\theta_l = dx \cdot \theta_0 - \theta_r.$$

We know that bigger φ brings bigger accuracy of the reconstruction process (see the experiment in Sec. 2.8.1). And since we would like to achieve the best accuracy

possible, we take bigger φ from the two possible values (φ_l and φ_r) and associated θ and calculate the depth estimation using Eq. (3.1).

Note that the vertical reconstruction is addressed in the same way as in Sec. 2.6.

The same reconstruction principle could also be used when we do the reconstruction from the non-symmetric pair of panoramas build from only one pixel column of the captured images. The only difference in this case is that φ_l and φ_r take only the value given by the pixel column that contributes to each panoramic image. The experiment in Sec. 3.6.1 proves the correctness of the principle.

3.5 Epipolar constraint

In the previous section we assumed that we can calculate the angle θ using Eq. (3.2). This equation holds if we do the reconstruction based on a symmetric pair of stereo panoramic images, which are made from one pixel column of the captured image. In this case we know that the epipolar lines are corresponding rows of the panoramic image (Sec. 2.5).

The stripe is composed of columns, each of them with a different angle φ. This basically means that we are dealing in fact with non-symmetric cases, for which the epipolar lines are different from corresponding rows. But if we look at the situation from another viewpoint, we can establish the following: We are using symmetric stripes to build a stereo pair of panoramic images. If we lower the resolution of the captured image, we transform the stripe into a column. The symmetric stripes would become symmetric columns and we could again use the rows of the panoramic image as epipolar lines. The same conclusion can be drawn from the property of the viewing circle, which gets thicker if we use a stripe instead of a column.

We can see this fact already by observing the light rays forming the column (Fig. 3.5) or the stripe (Fig. 3.3) that contributes to the panoramic image. In both examples the viewing circle should be thicker as presented, because the column and the stripe are formed from the set of light rays, which intersect in the optical center of the camera.

Lower resolution also brings considerable decrease in the number of possible depth estimates (Sec. 2.7.2), so the quality of results obtained from stripes should be better than the quality of results obtained from columns captured at a suitable lower resolution. For illustration: from around 140 possible estimates for *camera #1* in Sec. 2.8 to only around 10 estimates for the same camera at 14 times lower resolution – the width of the stripe that is transformed into a column W_s=14. It is also much harder to find corresponding points in so low resolution images.

One very interesting property of the system is also that there is a possibility of reconstructing the scene from non-symmetric pairs of panoramas while still using the simple, horizontal epipolar geometry (see the experiment in Sec. 3.6.1).

3.6 Experimental results

Note that the basics about how the results are obtained, evaluated etc. are explained in Sec. 2.8.

The difference that should be exposed at this point is the modification of camera parameters, which are different due to building panoramic images from stripes (Sec. 3.4). The modified parameters are set to the following values, according to the camera, the width of the stripe W_s and the width of the captured images W (the width is 160 pixels in all experiments) that were used in the experiments:

- *camera #1*:
  - $2\varphi_{\max} = 29.75°$
  - $2\varphi_{\min} = 2\varphi_{\max} - 2(W_s - 1)\frac{\alpha}{W}$
  - $\theta_0 = W_s \cdot 0.205714°$
- *camera #2*:
  - $2\varphi_{\max} = 34.755°$
  - $2\varphi_{\min} = 2\varphi_{\max} - 2(W_s - 1)\frac{\alpha}{W}$
  - $\theta_0 = W_s \cdot 0.257143°$
- *camera #3*:
  - $2\varphi_{\max} = 14.46375°$
  - $2\varphi_{\min} = 2\varphi_{\max} - 2(W_s - 1)\frac{\alpha}{W}$
  - $\theta_0 = W_s \cdot 0.102857°$.

The old parameters presented in Sec. 2.8 have been used only when $W_s = 1$, if not stated otherwise.

These values ensure similar number of possible depth estimates (Sec. 2.7.7).

The real time sensor has been simulated by rotating one standard camera for the angle determined by the width of the stripe W_s.

3.6.1 Reconstruction from non-symmetric pairs of panoramas

Experiment background: In Sec. 3.4 we have introduced the principle for reconstruction when we use symmetric stripes instead of symmetric columns. We have mentioned that the same principle could also be used when we do the reconstruction from the non-symmetric pairs of panoramas build from only one pixel column ($W_s = 1$) of the captured images.

And in Sec. 3.5 we have stated that we can still use the simple, horizontal epipolar geometry in some cases. This cases have been investigated by Shum and Szeliski [20] and they have concluded that even in the non-symmetric cases the epipolar geometry is sufficiently close to the horizontal epipolar geometry if either r/l or φ are small. They have even presented the approximate numbers: $r/l \leq 0.\overline{6}$, $2\varphi \leq 30°$. In the example given here, we satisfied both criteria. Compare the panoramic images in Fig. 2.4. For more details see [20].

In our case in this section the corresponding points lie in the same image row determined by the epipolar geometry. The results were obtained with *camera #1*.
Results: The comparison of results obtained by processing symmetric pair of panoramas and non-symmetric pair of panoramas is presented in Tab. 3.1.
Conclusion: We can see that the results are similar. Thus, the experiment confirms that the suggested reconstruction principle is correct.

feature	d [cm]	symmetric pair $l(\alpha,\beta)$ [cm]	symmetric pair $l(\alpha,\beta)-d$ [cm (% of d)]	non-symmetric pair $l(\alpha,\beta)$ [cm]	non-symmetric pair $l(\alpha,\beta)-d$ [cm (% of d)]
1	165.0	162.3	-2.7 (-1.6%)	155.3	-9.7 (-5.9%)
2	119.0	118.0	-1.0 (-0.9%)	112.1	-6.9 (-5.8%)
3	128.0	133.7	5.7 (4.4%)	131.4	3.4 (2.7%)
4	126.5	125.6	-0.9 (-0.7%)	124.4	-2.1 (-1.7%)
5	143.0	146.7	3.7 (2.6%)	146.8	3.8 (2.6%)
6	143.0	151.9	8.9 (6.2%)	146.8	3.8 (2.7%)
7	142.5	152.7	10.2 (7.2%)	147.6	5.1 (3.6%)
8	136.5	141.0	4.5 (3.3%)	137.6	1.1 (0.8%)
9	104.5	106.8	2.3 (2.2%)	103.0	-1.5 (-1.4%)
10	81.7	79.6	-2.1 (-2.5%)	79.7	-2.0 (-2.5%)
11	84.5	80.6	-3.9 (-4.6%)	81.9	-2.6 (-3.1%)
12	83.5	82.7	-0.8 (-0.9%)	82.4	-1.1 (-1.3%)
13	97.0	94.9	-2.1 (-2.2%)	91.2	-5.8 (-6.0%)
14	110.0	114.9	4.9 (4.5%)	107.3	-2.7 (-2.5%)
15	180.0	191.1	11.1 (6.2%)	192.4	12.4 (6.9%)
16	124.5	129.9	5.4 (4.3%)	124.9	0.4 (0.3%)
17	132.5	132.4	-0.1 (-0.1%)	130.2	-2.3 (-1.8%)
18	134.5	136.6	2.1 (1.5%)	130.2	-4.3 (-3.2%)
19	113.0	109.4	-3.6 (-3.2%)	107.5	-5.5 (-4.8%)
20	125.0	121.6	-3.4 (-2.8%)	117.8	-7.2 (-5.7%)
21	130.0	128.8	-1.2 (-1.0%)	123.8	-6.2 (-4.8%)
		$\text{AVG}_{\%}$=3% ± 2%		$\text{AVG}_{\%}$=3.3% ± 1.9%	

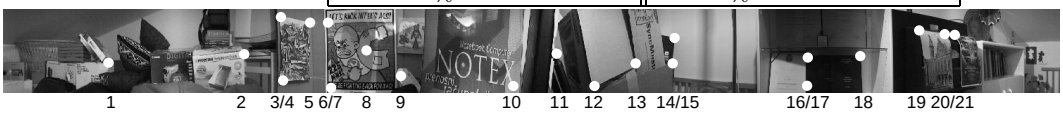


Table 3.1: The results obtained by processing symmetric pair of panoramas ($2\varphi = 29.9625°$; the same as in Tab. 2.11 for *camera #1*) and non-symmetric pair of panoramas ($2\varphi_l = 29.9625°$, $2\varphi_r = 3.6125°$).

3.6.2 Reconstruction from stripe panoramas

Experiment background: We want to determine the influence of the width of the stripe (W_s) that contributes to the panoramic image on the reconstruction accuracy ($\text{AVG}_{\%}$).

The results were obtained with *camera #1.*

Results: The results for four different widths of the stripes (W_s) are given in Tab. 3.2. Note that the width of the captured images from which the panoramic images have been constructed is 160 pixels.

feature	d [cm]	W_s=2 $l(\alpha,\beta)$ [cm]	W_s=2 $l(\alpha,\beta)-d$ [cm (% of d)]	W_s=6 $l(\alpha,\beta)$ [cm]	W_s=6 $l(\alpha,\beta)-d$ [cm (% of d)]	W_s=14 $l(\alpha,\beta)$ [cm]	W_s=14 $l(\alpha,\beta)-d$ [cm (% of d)]	W_s=29 $l(\alpha,\beta)$ [cm]	W_s=29 $l(\alpha,\beta)-d$ [cm (% of d)]
1	165.0	165.8	0.8 (0.5%)	162.9	-2.1 (-1.3%)	130.1	-34.9 (-21.1%)	96.3	-68.7 (-41.6%)
2	119.0	114.1	-4.9 (-4.1%)	103.2	-15.8 (-13.3%)	117.4	-1.6 (-1.3%)	113.0	-6.0 (-5.0%)
3	128.0	136.3	8.3 (6.5%)	141.1	13.1 (10.2%)	133.0	5.0 (3.9%)	163.4	35.4 (27.7%)
4	126.5	127.7	1.2 (0.9%)	119.0	-7.5 (-5.9%)	132.5	6.0 (4.8%)	128.9	2.4 (1.9%)
5	143.0	145.0	2.0 (1.4%)	143.3	0.3 (0.2%)	117.6	-25.4 (-17.8%)	225.2	82.2 (57.5%)
6	143.0	154.7	11.7 (8.2%)	143.3	0.3 (0.2%)	117.5	-25.5 (-17.8%)	89.7	-53.3 (-37.3%)
7	142.5	155.5	13.0 (9.1%)	161.7	19.2 (13.5%)	179.3	36.8 (25.8%)	179.8	37.3 (26.2%)
8	136.5	144.0	7.5 (5.5%)	126.9	-9.6 (-7.0%)	111.3	-25.2 (-18.5%)	177.4	40.9 (29.9%)
9	104.5	103.5	-1.0 (-0.9%)	101.6	-2.9 (-2.8%)	112.8	8.3 (7.9%)	97.1	-7.4 (-7.1%)
10	81.7	80.1	-1.6 (-2.0%)	76.7	-5.0 (-6.1%)	75.4	-6.3 (-7.7%)	57.3	-24.4 (-29.9%)
11	84.5	79.6	-4.9 (-5.8%)	81.1	-3.4 (-4.0%)	74.9	-9.6 (-11.3%)	55.0	-29.5 (-34.9%)
12	83.5	83.3	-0.2 (-0.3%)	81.6	-1.9 (-2.2%)	83.2	-0.3 (-0.3%)	58.8	-24.7 (-29.6%)
13	97.0	97.4	0.4 (0.4%)	95.5	-1.5 (-1.6%)	78.6	-18.4 (-19.0%)	74.1	-22.9 (-23.6%)
14	110.0	113.6	3.6 (3.2%)	107.8	-2.2 (-2.0%)	117.6	7.6 (6.9%)	101.7	-8.3 (-7.6%)
15	180.0	182.0	2.0 (1.1%)	165.9	-14.1 (-7.8%)	203.2	23.2 (12.9%)	96.4	-83.6 (-46.4%)
16	124.5	128.2	3.7 (3.0%)	125.8	1.3 (1.0%)	142.0	17.5 (14.1%)	138.6	14.1 (11.3%)
17	132.5	127.7	-4.8 (-3.6%)	131.6	-0.9 (-0.7%)	150.8	18.3 (13.8%)	148.2	15.7 (11.9%)
18	134.5	134.9	0.4 (0.3%)	131.6	-2.9 (-2.1%)	97.8	-36.7 (-27.3%)	119.7	-14.8 (-11.0%)
19	113.0	113.1	0.1 (0.1%)	111.0	-2.0 (-1.8%)	101.9	-11.1 (-9.8%)	97.1	-15.9 (-14.1%)
20	125.0	126.5	1.5 (1.2%)	118.6	-6.4 (-5.1%)	124.5	-0.5 (-0.4%)	120.0	-5.0 (-4.0%)
21	130.0	127.2	-2.8 (-2.2%)	137.4	7.4 (5.7%)	140.9	10.9 (8.4%)	137.8	7.8 (6.0%)
		$\text{AVG}_{\%}$=2.9% ± 2.7%		$\text{AVG}_{\%}$=4.5% ± 4%		$\text{AVG}_{\%}$=11.9% ± 7.9%		$\text{AVG}_{\%}$=22.1% ± 15.8%	

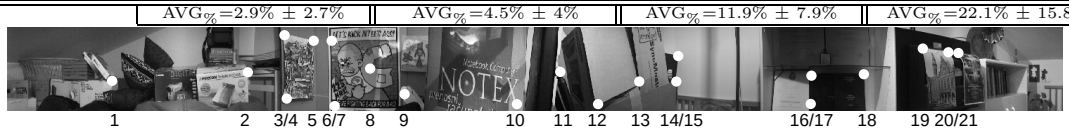


Table 3.2: The results obtained with four different widths of the stripes (W_s).

Conclusion: As expected, the accuracy deteriorates with wider stripes. Though, if we compare the results obtained with W_s=2 and W_s=6 with the results obtained with W_s=1 (e.g. Tab. 3.1), we see that the results are still very good if not very similar.

The graph in Fig. 3.11 shows that the relation between $\text{AVG}_{\%}$ and W_s ($\text{AVG}_{\%}(W_s)$) can be well approximated with a linear function, although we have fitted a second-order polynomial on the (sparse) data.

Another fact is that with the wider stripe more scene points are not captured in the panoramic images (Sec. 3.2.1), which means that the number of pixels in the left eye panoramic image that do not have the corresponding point in the right eye panoramic image increases. So, the generated panoramic images differ more and more from the ideal case (W_s=1) as we are increasing the width of the stripe W_s. In the ideal case each scene point in the left eye panoramic image has its

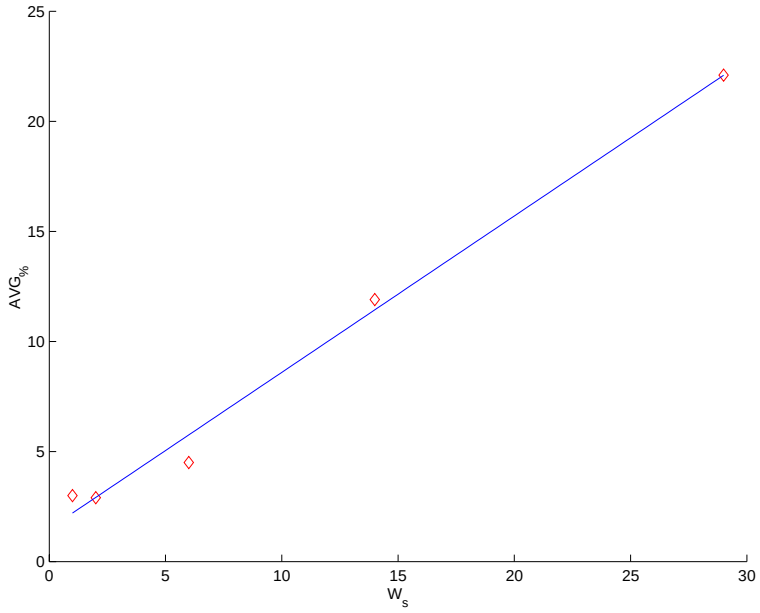


Figure 3.11: The relation between $\mathrm{AVG}_{\%}$ and W_s ($\mathrm{AVG}_{\%}(W_s)$) can be well approximated with a linear function. The diamonds represent the data involved in the approximation.

corresponding point in the right eye panoramic image; exceptions are of course related to occlusions. – We performed an experiment in which we used *camera #1* (Sec. 2.8, W_s=1) to capture a banner written all over with text of different sizes, styles and fonts positioned at approximately 2 cm from the camera lens (on each location of the camera on the camera path). We were unable to prove from the human perceptual point of view the hypothesis that some scene points in this case are not captured. Really small deviations of the generated panoramic images from the real banner could as well be attributed to the fact that the images are discrete (for example: we cannot take a half of a pixel). On the other hand, the hypothesis is confirmed if stripes ($W_s > 1$) are used (Figs. 3.1 and 3.2).

3.6.3 Reconstruction from stripe panoramas — Different room

Experiment background: We want to see if we can achieve similar results as in Sec. 3.6.2, using the same camera (*camera #1*) in a different room?
Results: The results obtained for two most interesting stripe widths (according to Tab. 3.2) in the different room are presented in Tab. 3.3.

feature	d [cm]	W_s=6 $l(\alpha,\beta)$ [cm]	W_s=6 $l(\alpha,\beta)-d$ [cm (% of d)]	W_s=14 $l(\alpha,\beta)$ [cm]	W_s=14 $l(\alpha,\beta)-d$ [cm (% of d)]
1	63.2	59.0	-4.2 (-6.6%)	57.0	-6.2 (-9.8%)
2	51.5	48.8	-2.7 (-5.3%)	54.3	2.8 (5.5%)
3	141.0	161.3	20.3 (14.4%)	196.7	55.7 (39.5%)
4	142.0	143.6	1.6 (1.1%)	246.6	104.6 (73.7%)
5	216.0	213.3	-2.7 (-1.2%)	225.7	9.7 (4.5%)
6	180.0	209.2	29.2 (16.2%)	184.9	4.9 (2.7%)
7	212.0	200.4	-11.6 (-5.5%)	167.0	-45.0 (-21.2%)
8	49.0	44.8	-4.2 (-8.6%)	48.6	-0.4 (-0.9%)
9	49.0	45.5	-3.5 (-7.2%)	45.5	-3.5 (-7.2%)
10	97.0	95.7	-1.3 (-1.3%)	90.6	-6.4 (-6.6%)
11	129.5	132.6	3.1 (2.4%)	102.1	-27.4 (-21.1%)
12	134.0	139.8	5.8 (4.3%)	162.8	28.8 (21.5%)
13	119.0	122.7	3.7 (3.1%)	117.9	-1.1 (-0.9%)
14	156.0	163.0	7.0 (4.5%)	138.2	-17.8 (-11.4%)
15	91.0	94.0	3.0 (3.3%)	81.7	-9.3 (-10.2%)
16	97.7	99.6	1.9 (1.9%)	96.9	-0.8 (-0.8%)
17	111.0	110.9	-0.1 (-0.1%)	169.4	58.4 (52.6%)
18	171.5	191.1	19.6 (11.4%)	147.3	-24.2 (-14.1%)
19	171.5	165.8	-5.7 (-3.3%)	170.4	-1.1 (-0.6%)
		AVG$_{\%}$=5.4% ± 4.5%		AVG$_{\%}$=16% ± 19.6%	

Table 3.3: The results obtained in the different room, but with the same camera as in Sec. 3.6.2.

Conclusion: The overall results are a bit different, especially the results for W_s=14, though still similar. But this was also expected since the results for W_s=1 (Tab. 2.10) are also a bit worse compared to the results obtained in a previous room (the symmetric pair in Tab. 3.1). As already mentioned in Sec. 2.8.7, small differences in

results are expected, since each room has its own shape, i.e. the depth distribution around the center of the system is different. And we know how this influences the accuracy, while we are limited with the number of possible depth estimates, which are approximations of the real distances (Sec. 2.7.4).

3.6.4 Reconstruction from stripe panoramas — Different cameras

Experiment background: We want to see how is the reconstruction error related to different cameras used in the same room, when the width of the stripe is constant for all cameras? As written in Sec. 3.6 the results were obtained at similar number of possible depth estimates.

Results: The comparison of results for the width of the stripe W_s=14 for three different cameras is given in Tab. 3.4. Note that for features marked 3, 5, 6, 7, 15, 19, 20 and 21 the real distance d in case of *camera #3* is different from the presented one. The reason for this lies in the vertical view angle of the camera β, which is smaller in comparison to other two cameras. This means that some marked feature points are not seen in the panoramic images generated with *camera #3*, so we have chosen a nearby features with similar distances. By all means, in the calculations we have used the correct distances.

feature	d [cm]	*camera #1* $l(\alpha,\beta)$ [cm]	*camera #1* $l(\alpha,\beta)-d$ [cm (% of d)]	*camera #2* $l(\alpha,\beta)$ [cm]	*camera #2* $l(\alpha,\beta)-d$ [cm (% of d)]	*camera #3* $l(\alpha,\beta)$ [cm]	*camera #3* $l(\alpha,\beta)-d$ [cm (% of d)]
1	165.0	130.1	-34.9 (-21.1%)	178.7	13.7 (8.3%)	144.6	-20.4 (-12.3%)
2	119.0	117.4	-1.6 (-1.3%)	167.0	48.0 (40.4%)	130.0	11.0 (9.3%)
3	128.0	133.0	5.0 (3.9%)	120.9	-7.1 (-5.5%)	118.1	-7.4 (-5.9%)
4	126.5	132.5	6.0 (4.8%)	99.0	-27.5 (-21.7%)	95.5	-31.0 (-24.5%)
5	143.0	117.6	-25.4 (-17.8%)	127.4	-15.6 (-10.9%)	124.6	-16.4 (-11.6%)
6	143.0	117.5	-25.5 (-17.8%)	231.5	88.5 (61.9%)	114.3	-27.2 (-19.2%)
7	142.5	179.3	36.8 (25.8%)	134.8	-7.7 (-5.4%)	154.5	12.5 (8.8%)
8	136.5	111.3	-25.2 (-18.5%)	183.0	46.5 (34.1%)	119.2	-17.3 (-12.7%)
9	104.5	112.8	8.3 (7.9%)	106.7	2.2 (2.1%)	96.8	-7.7 (-7.4%)
10	81.7	75.4	-6.3 (-7.7%)	95.9	14.2 (17.4%)	78.8	-2.9 (-3.5%)
11	84.5	74.9	-9.6 (-11.3%)	69.5	-15.0 (-17.7%)	88.8	4.3 (5.0%)
12	83.5	83.2	-0.3 (-0.3%)	69.7	-13.8 (-16.5%)	75.1	-8.4 (-10.0%)
13	97.0	78.6	-18.4 (-19.0%)	93.3	-3.7 (-3.8%)	102.3	5.3 (5.5%)
14	110.0	117.6	7.6 (6.9%)	98.8	-11.2 (-10.2%)	108.2	-1.8 (-1.6%)
15	180.0	203.2	23.2 (12.9%)	197.1	17.1 (9.5%)	174.8	3.8 (2.2%)
16	124.5	142.0	17.5 (14.1%)	108.3	-16.2 (-13.0%)	118.1	-6.4 (-5.2%)
17	132.5	150.8	18.3 (13.8%)	167.0	34.5 (26.0%)	123.6	-8.9 (-6.7%)
18	134.5	97.8	-36.7 (-27.3%)	153.9	19.4 (14.5%)	102.0	-32.5 (-24.2%)
19	113.0	101.9	-11.1 (-9.8%)	126.3	13.3 (11.8%)	100.4	-7.6 (-7.0%)
20	125.0	124.5	-0.5 (-0.4%)	168.3	43.3 (34.6%)	129.9	10.9 (9.2%)
21	130.0	140.9	10.9 (8.4%)	183.8	53.8 (41.4%)	102.1	-22.9 (-18.3%)
		$AVG_{\%}$=11.9% ± 7.9%		$AVG_{\%}$=19.4% ± 15.3%		$AVG_{\%}$=10% ± 6.6%	

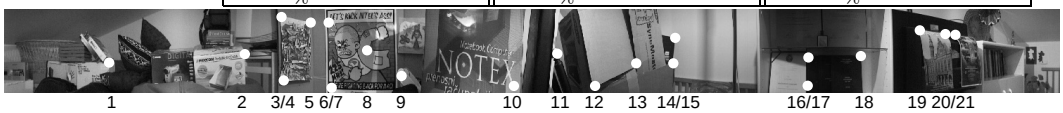


Table 3.4: The results obtained with three different cameras, while the width of the stripe W_s=14.

Conclusion: The results show that with bigger horizontal view angle α and at

Figure 3.12: Parts of panoramic images from distorted (left) and undistorted (right) sequence of images. We can see that the problem with missing scene points in the panoramic images (Sec. 3.2.1) remains. Compare the images with the image presented in the bottom of Tab. 3.4.

constant width of the stripe W_s (> 1) the reconstruction accuracy deteriorates:

$$\alpha_{camera\#3} < \alpha_{camera\#1} < \alpha_{camera\#2}$$
$$\Longleftrightarrow$$
$$\mathrm{AVG}\%_{camera\#3} < \mathrm{AVG}\%_{camera\#1} < \mathrm{AVG}\%_{camera\#2}.$$

The result in Tab. 3.3 for W_s=14, gained with *camera #1*, but in the different room, also satisfies this relation.

Bigger horizontal view angle α means that the lens distortion in captured images is more obvious (Sec. 2.8.10). Thus, this fact in relation with the above conclusion suggests that by using undistorted images, we could obtain better reconstruction accuracy. But the overall reconstruction accuracy obtained using undistorted sequence is very similar to that obtained using distorted sequence. This is due to the fact that we can see that the problem with missing scene points in the panoramic images (Sec. 3.2.1) remains already by looking at built panoramic images from distorted and undistorted sequence of images (Fig. 3.12).

So, one of the reasons for the lower reconstruction accuracy is again the fact that there are parts of the scene which are not captured in the panoramic image. This means that the number of pixels in the left eye panoramic image that do not have the corresponding point in the right eye panoramic image increases with bigger horizontal view angle α at constant width of the stripe W_s, while the shift angle θ_0 between two successively positioned cameras is bigger for bigger α (Sec. 3.6).

In Sec. 2.8.8 we have concluded that similar overall accuracy can be achieved if we use different cameras, while the width of the stripe W_s=1. If we relate this conclusion to the main conclusion of this section (with bigger horizontal view angle α and at constant width of the stripe W_s (> 1) the reconstruction accuracy deteriorates), we can conclude that the approximation presented in Fig. 3.11 gets steeper with increasing horizontal view angle α.

3.7 Summary

This part of the thesis evaluated our idea for the real time extension of the basic mosaic-based panoramic depth imaging system.

The presented theory and results suggest that the new sensor could be used for real time capturing of panoramic depth images and consequently for autonomous navigation of a mobile robot in a room. But this statement is unfortunately highly related to the application of the system and its demanded reconstruction accuracy. If in the first part of the thesis we could conclude that by all means the system can be used for robot localization and navigation in a room, we are limited with the number of constraints when it comes to the real time sensor. On the other hand, assumptions made in this part of the thesis have proved to be correct and revealed some other interesting properties of the system.

The main conclusions made in this chapter are summarized in Sec. 4.2.

Chapter 4

Conclusion

4.1 Dissertation summary

In this dissertation we have designed, implemented, tested and evaluated our panoramic depth imaging system from multiperspective panoramic images. After analyzing the basic system, which is based only on one standard camera that is offset from the system's rotational center, we have suggested a real time extension of the system, based on simultaneously using many standard cameras. We have not physically built the real time sensor, but we have performed simulations to establish the quality of results. Both systems have been exhaustively analyzed and compared. The analyses revealed a number of interesting properties of the systems, which are summarized in the next section.

Theoretically, this study reveals relations between system parameters, speed of stereo images acquisition, speed of reconstruction, reconstruction accuracy and 3D scene of interest. In practice, the study can serve as a guide for panoramic depth imaging sensor design, or as a guide for designing interactive user interface for determining system parameters based on our needs and also as a framework for autonomous robot localization and navigation system.

Our basic goal in this dissertation was to establish whether our system can be used for robot localization and navigation tasks. In order to do that, we have analyzed the system: According to the basic system accuracy we definitely can use it for such tasks. Because the time needed to capture stereo images in the basic system is far from real time, we have upgraded the basic system in this direction. The assumptions made in this extension have been proved to be correct, but the accuracy of the new sensor generally deteriorates in comparison to the basic sensor.

4.2 Conclusions

In this section we summarize the main conclusions made throughout the dissertation and indicate in which section of the dissertation each conclusion has been made.

About the basic system (Chapter 2):

- The geometry of capturing multiperspective panoramic images can be described with a pair of parameters (r, φ) (Sec. 2.4). By increasing (decreasing) each of them, we increase (decrease) the baseline $(2r_0)$ of our stereo system.

- The stereo pair acquisition procedure with only one standard camera cannot be executed in real time (Sec. 2.7.1).

- The epipolar geometry in case of symmetric stereo pair of panoramic images, which we use in the reconstruction process, is very simple: epipolar lines are image rows (Sec. 2.5).

- The parameters of the system should be estimated as precisely as possible, since already a small difference can cause a big difference in the reconstruction accuracy of the system (Secs. 2.7.2, 2.8.5, 2.8.6 and 2.8.9).

- We can effectively constrain the search space on the epipolar line (Sec. 2.7.3). This follows directly from the interpretation of the equation for depth estimation l (Eq. (2.5)), while other rules for constraining the search space, known from traditional stereo vision systems, can also be applied in addition to the basic constraint. An example of such rule is to seek for the neighboring pair of corresponding points only from the previously found correspondence on.

- The confidence in the estimated depth is variable: 1) the bigger the slope of the function l, the smaller the confidence in the estimated depth (one-pixel error Δl gets bigger) and 2) the bigger the value φ for each camera (α), the bigger the number of possible depth estimates and consequently the bigger the confidence (Secs. 2.7.4, 2.8.1 and 2.8.3).

- We can influence the parameter θ_0 by varying the resolution of captured images or by varying the horizontal view angle α (Secs. 2.7.4 and 2.7.7).

- By varying the radius r, we vary the biggest possible and sensible depth estimation l and the size of the one-pixel error Δl (Sec. 2.7.4).

- The bigger the value α, the smaller the horizontal resolution of panoramic images at fixed resolution of captured images (Sec. 2.7.7). Consequently, the number of possible depth estimates per one degree gets lower.

- In practice, from the autonomous robot localization and navigation system point of view, we should define the upper boundary of the allowed one-pixel error size Δl (Sec. 2.7.5).

- The contribution of the vertical reconstruction is small in general, but has a positive influence on the overall results (Secs. 2.7.6 and 2.8.4).

- The numbers of possible depth estimates are very similar for different cameras (α) at fixed resolution of the captured images (Sec. 2.7.7).

- The size of the one-pixel error Δl is also similar at similar number of possible depth estimates for different cameras (Sec. 2.7.7).

- The reconstruction process can execute in real time (Sec. 2.8.2).

- The reconstructed points lie on concentric circles centered in the center of rotation and the distance between circles (the one-pixel error Δl) increases the further away they lie from the center (Secs. 2.7.4 and 2.8.3).

- The linear model for estimation of angle φ have been proved better for a given set of parameters in comparison to the non-linear model (Sec. 2.8.6).
- We can achieve similar reconstruction accuracy with panoramas build from only one-pixel column ($W_s = 1$) of the captured images in different rooms, even with different cameras (Secs. 2.8.7 and 2.8.8).
- The remaining error in accuracy could be attributed to a number of possible reasons (Sec. 2.8.9).
- Processing undistorted images in general brings better though comparable results, but undistorting the sequence can be time expensive task and we are forced to re-estimate some parameters of the system after the distortion is corrected (Sec. 2.8.10).

About the real time system (Chapter 3):

- Building panoramic images from wider stripes than only one-pixel column brings faster execution of the building process and worse quality of the generated panoramas (Sec. 3.2).
- The stripes that contribute to the panoramic image do not cover all the scene (Sec. 3.2.1). The wider is the stripe, more scene points are not captured in the panoramic image (the number of points without correspondences increases (Sec. 3.6.2)). By using smaller r the gaps are smaller, which means that with smaller r we cover more scene points. The same goes for different values of α. This holds also for one-pixel column stripes.
- If we want to capture the parallax effect, which is essential for the reconstruction, we have to accept the fact that not all the scene points are captured (Sec. 3.2.1).
- If we could have a camera on each position on the circular path, we could build the panoramic image in real time and achieve the same quality of results as in case of the basic system, but in practice we should make a compromise between the width of the stripes, the number of cameras in respect to their parameters (e.g. α and external dimensions), the value of radius r and the needed accuracy of the system (Sec. 3.3).
- The stereo reconstruction procedures along with the epipolar geometry have been proved correct for both investigated cases: the basic sensor and the real time sensor (Secs. 3.4, 3.5 and 3.6).
- Using images with resolution 160×120 represents a good compromise between overall time complexity of the system and its accuracy (Secs. 2.7.1, 2.8.2 and 3.3).

- Even in some cases of non-symmetric pairs of panoramas we can use our reconstruction procedure (Sec. 3.6.1).
- The reconstruction accuracy deteriorates approximately linearly with wider stripes (Sec. 3.6.2).
- We can achieve similar reconstruction accuracy with panoramas build from stripes ($W_s > 1$) with fixed size W_s of the captured images in different rooms at fixed horizontal view angle α (Sec. 3.6.3), but with bigger horizontal view angle α and at constant width of the stripe W_s (> 1) the reconstruction accuracy deteriorates (Sec. 3.6.4).
- In general, by undistorting the image sequence and building panoramas from stripes ($W_s > 1$), we do not obtain better reconstruction accuracy (Sec. 3.6.4).
- The approximate linear dependency of accuracy on the width of the stripes gets steeper with increasing horizontal view angle α (Sec. 3.6.4).

All this is true for the cameras used in the dissertation, while for really wide angle cameras some conclusions perhaps demand further investigation in direction presented by the conclusion (see the last section).

We should also expose the fact that we have developed few other simple procedures along the way, which have been proved useful in various aspects. Like the method for estimating the position of the optical center (Sec. 2.7.2) or the method for defining the maximal reliable depth value (Sec. 2.7.5).

The main conclusion in respect to the robot localization and navigation tasks has already been written in the last paragraph in the previous section.

4.3 Contributions to science

We have evaluated the system from the robot localization and navigation point of view, so some conclusions are more or less specific (like the definition of the maximal reliable depth value (Sec. 2.7.5)). Some conclusions are known from work of other authors, but a number of them are new in sense that they have not been published by others [38, 47, 58]. In this respect the main contributions are:

- We have proved that constraining the search space on the epipolar line follows directly from the equation for depth estimation (Sec. 2.7.3). This means that we have to search for the corresponding point only on the part of the epipolar line, while its remaining parts cannot contain the corresponding point at all.
- We have proved that the confidence in the estimated depth is variable (Secs. 2.7.4, 2.8.1 and 2.8.3). This is true not only for the case known from traditional stereo, where with bigger distance the one-pixel error in disparity estimation

also gets bigger, but also for the case where with bigger φ for each camera (α) the confidence increases due to a bigger number of possible depth estimates.

- We have investigated the influence of different cameras (α) on the overall system performance and not only on its accuracy (Sec. 2.7.7). Each α gives a specific number of possible depth estimates per one degree.
- We have suggested and evaluated a real time extension of the basic system (Chapter 3). We have tested the idea similar to that for taking the omnidirectional depth information by sonar sensors, but instead of sonars we have used standard cameras.
- We have shown that the columns or the stripes that contribute to the stereo panoramas do not cover all the scene (Sec. 3.2.1).
- We have shown that the reconstruction accuracy deteriorates approximately linearly with wider stripes (Sec. 3.6.2) as well as that with bigger horizontal view angle α and at constant width of the stripe W_s (> 1) the reconstruction accuracy also deteriorates (Sec. 3.6.4).
- We have investigated the impact of lens distortion on the reconstruction accuracy (Secs. 2.8.10 and 3.6.4).
- And last, but not least, the dissertation can serve as a guide for panoramic depth imaging sensor design and related issues.

4.4 Future work

There are a lot of details that could be further investigated, but the global picture is already here. For instance, we could investigate what is the contribution of the non-linear model for estimation of angle φ (Sec. 2.8.6), if we use a really wide angle camera. The influence of lens distortion (Secs. 2.8.10 and 3.6.4) in such case would also clarify the global picture. In this dissertation this is not done because of the lack of equipment.

Then there are technical problems that should be addressed properly. Such problems could appear at building the real, physical real time sensor and real time implementation.

On the other hand, we could estimate how good are the results, gained with a good algorithm for searching the correspondences, for the autonomous robot localization and navigation. Perhaps even by using the image pyramids, the multiple-baseline stereo approach and/or the sub-pixel accuracy procedure.

Different authors have already addressed these (e.g. a multiple-baseline stereo) and other (e.g. a generalization of the epipolar geometry) sub-problems, so one could also do a generalization for the autonomous robot localization and navigation,

though it would introduce more complex models, which demand more processing power.

The use of multiperspective panoramas is also studied for visualization purposes, viewable stereo, new-view synthesis etc.

Appendix A

Extended Abstract in Slovenian Language

Razširjen povzetek:

Gradnja globinskih panoramskih slik v realnem času z uporabo standardnih kamer

A.1 Uvod

A.1.1 Opis ožjega znanstvenega področja

Disertacija spada na področje računalniških znanosti, na področje računalniškega vida. Osnovna ideja računalniškega vida je interpretacija slikovnega materiala. Del tega področja se ukvarja z rekonstrukcijo 3D informacij iz slik. Eden izmed osnovnih pristopov k 3D rekonstrukciji je stereo vid. Pod pojmom stereo rekonstrukcija v osnovi razumemo gradnjo globinskih slik iz dveh zajetih slik. Vsaka globinska slika hrani ocene razdalj do objekta iz enega gledišča. V disertaciji se osredotočimo na panoramske slike. To so slike, ki obsegajo večji zorni kot kot nam ga dajo standardne kamere, celo večji od človeškega vidnega polja.

A.1.2 Opis problema

Glavni problem, ki bi ga radi rešili v tej disertaciji, predstavlja določitev in analizo lastnosti ter učinkovitosti globinskega panoramskega senzorja, ki temelji na večperspektivnih panoramskih slikah. Pomembna lastnost (dveh ali več) večperspektivnih panoram je, da hranijo informacijo o paralaksi gibanja, saj so kolone le-teh zajete iz različnih perspektiv. Analiza učinkovitosti bo podala odgovor na vprašanje, ali so rezultati uporabni za samodejno lokalizacijo in navigacijo mobilnega robota. Za gradnjo sistema je uporabljena le standardna oprema. Nadgradnja delovanja osnovnega sistema, s ciljem doseči delovanje v realnem času, je simulirana z namenom določitve učinkovitosti novega senzorja v primerjavi z osnovnim.

A.1.3 Zgradba disertacije

V tem poglavju smo orisali problem, ki ga rešuje disertacija. V naslednjem poglavju obravnavamo osnovni sistem gradnje globinskih panoramskih slik, ki temelji na vrtenju standardne kamere. V poglavju A.3 predlagamo nadgradnjo osnovnega sistema in ocenimo učinkovitost rešitve za delovanje v realnem času. Zaključek z izpostavljenimi prispevki k znanosti je podan v zadnjem poglavju.

A.2 Osnovni sistem

A.2.1 Uvod

Na sliki A.1 so na desni sliki prikazani strojni deli našega sistema: rotacijska roka, ki se vrti v horizontalni smeri, na njo je pritrjen distančnik, ki omogoča premik optičnega centra kamere iz središča vrtenja, na ta distančnik pa je pritrjena ena sama standardna kamera, ki gleda navzven, torej stran od središča vrtenja. Panoramske slike ustvarimo tako, da rotacijsko roko vrtimo za kot, ki ustreza premiku za

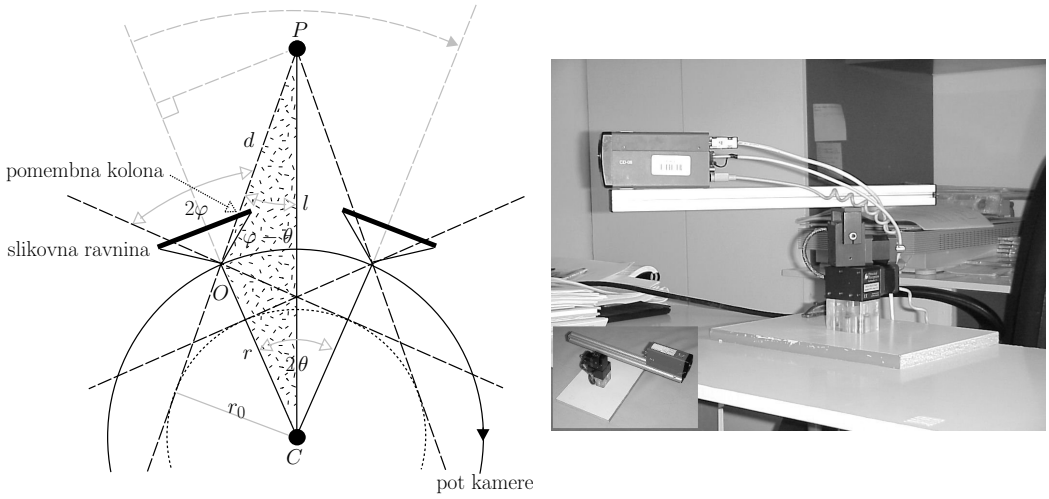


Figure A.1: Tloris geometrije sistema za ustvarjanje večperspektivnih panoram. Optična os kamere je v horizontalnem položaju. Na desni sliki so prikazani strojni deli sistema.

eno kolono slikovnih elementov zajete slike, te kolone pa mozaičimo v panoramsko sliko.

Stereo rekonstrukcijo naredimo na osnovi simetričnega para večperspektivnih panoram. Simetričen par panoram dobimo, če vzamemo enako oddaljeni koloni levo in desno od sredine zajete slike.

A.2.2 Sorodna dela

Večperspektivne panorame same po sebi niso nekaj novega [20]: so poseben primer *večperspektivnih panoram za animacijo celic* (ang. *multiperspective panoramas for cel animation*) [13], poseben primer *križne projekcije* (ang. *crossed-slits (X-slits) projection*) [53, 56, 61], so zelo podobne slikam narejenim s postopkom *večsrediščna projekcija* (ang. *multiple-center-of-projection*) [17], postopkom *mnogotera projekcija* (ang. *manifold projection*) [27] in postopkom *krožna projekcija* (ang. *circular projection*) [19, 26]. Princip je zelo podoben tudi postopku gradnje panorame z *linearno potisno kamero* (ang. *linear pushbroom camera*) [10].

V člankih, ki so najbližji našemu delu [1, 20, 21], smo pogrešali predvsem dve stvari: analizo 1) zmogljivosti sistema in 2) iskanja korespondenčnih točk z uporabo epipolarne omejitve, zato smo temu posvetili bistveno pozornost. Medtem ko v [1] iščejo korespondenčne točke s sledenjem značilke tako, da sledijo značilko iz kolone, ki gradi eno panoramo, do kolone, ki gradi drugo panoramo, v [20] uporabljajo nadgrajen postopek stereo s preslikavo na ravnino (ang. *plain sweep stereo*). Glavna

ideja v [21] je optimizacija vhoda za tradicionalne postopke iskanja korespondenčnih točk z namenom izboljšanja rezultatov.

A.2.3 Geometrija sistema

Iz slike A.1 je razvidna splošna geometrija našega sistema za ustvarjanje večperspektivnih panoramskih slik, ki služijo kot vhod za gradnjo globinskih panoramskih slik. Točka C predstavlja središče vrtenja, okoli katerega se vrti kamera na oddaljenosti r, ki predstavlja polmer krožnice (pot) po kateri potuje kamera. Kamera je obrnjena navzven, torej stran od središča vrtenja. Optični center kamere se nahaja v točki O. Izbrana kolona slikovnih elementov, ki bo doprinesla k večperspektivni panorami, vsebuje preslikavo točke P na sceni. Ta je od točke C oddaljena za razdaljo l, od točke O pa za razdaljo d. S θ je označen trenutni kot zasuka med daljicama, ki sta definirani s središčem vrtenja C in optičnim centrom O ter središčem vrtenja C in točko na sceni P. S φ pa je označen kot med daljicama, ki sta definirani z optičnim centrom O in sredinsko kolono zajete slike ter optičnim centrom O in izbrano kolono zajete slike, ki bo doprinesla k večperspektivni panorami. Na ta kot lahko gledamo tudi kot na zmanjšanje horizontalnega zornega kota kamere α zajete slike.

A.2.4 Epipolarna geometrija

Izkaže se, da je epipolarna geometrija našega sistema zelo enostavna v primeru računanja rekonstrukcije na podlagi simetričnega para panoram: epipolarne premice simetričnih parov panoram so istoležne vrstice panoram. Dokaz si lahko bralec ogleda v [20, 23, 35, 51].

A.2.5 Stereo rekonstrukcija

Vrnimo se k sliki A.1. Na osnovi trigonometričnih relacij moramo zapisati enačbo za oceno globine l točke P na sceni. Po sinusnem izreku velja (opazujemo pobarvan trikotnik na sliki A.1):

$$l = \frac{r \cdot \sin(180^\circ - \varphi)}{\sin(\varphi - \theta)} = \frac{r \cdot \sin\varphi}{\sin(\varphi - \theta)}. \tag{A.1}$$

Iz zgornje enačbe sledi, da lahko globino l ocenimo le, če poznamo tri parametre: r, φ in θ. r je dan. Kot φ izračunamo glede na horizontalni zorni kot kamere α ali glede na goriščno razdaljo f po enačbi:

$$2\varphi = \frac{\alpha}{W} \cdot W_{2\varphi} \text{ ali } \varphi = \arctan\frac{W_{2\varphi}/2}{f}, \tag{A.2}$$

pri čemer je W širina zajete slike v slikovnih elementih, $W_{2\varphi}$ pa širina slike med kolonoma, ki gradita simetrični par panoram, prav tako v slikovnih elementih. Za

izračun kota θ pa moramo najprej najti korespondenčni točki na panoramskih slikah. Naš sistem deluje tako, da kamero po krožni poti premikamo za kot, ki ustreza eni koloni zajete slike. Če ta kot označimo s θ_0, potem za kot θ velja:

$$\theta = dx \cdot \frac{\theta_0}{2}, \tag{A.3}$$

kjer je dx absolutna vrednost razlike med slikovnima koordinatama korespondenčnih točk na horizontalni osi x panoramskih slik.

Ker pa enačba (A.1) poda razdaljo le do pravokotne projekcije točke na sceni na ravnino vrtenja kamere, moramo ustrezno upoštevati tudi vertikalno rekonstrukcijo, torej vertikalni zorni kot kamere β ($l = l(\alpha)$):

$$l(\alpha, \beta) = \sqrt{l(\alpha)^2 + \left(\frac{l(\alpha) \cdot \sin\theta}{\sin\varphi} \cdot \tan\omega_1\right)^2}, \tag{A.4}$$

kjer ω_1 izračunamo po enakem principu kot v enačbi (A.2).

A.2.6 Analiza zmogljivosti sistema

Čas gradnje panoramskih slik

Za naš sistem velja, da zgradimo v enem obhodu okoli vertikalne osi sistema 11 panoram (pet simetričnih parov in panoramsko sliko iz sredinskih kolon zajetih slik). Zajamemo 1501 sliko ($\alpha = 34°$), ločljivosti 160×120 slikovnih elementov, razdalja $r = 30$ cm, kot premika rotacijske roke $\theta_0 = 0,205714°$ in za to porabimo dobrih 15 minut na računalniku PC Intel PII./350 MHz.

Vpliv osnovnih parametrov na rekonstrukcijsko točnost

Za $\alpha = 34°$ in $W = 160$ slikovnih elementov je $\theta_0 = \frac{\alpha}{W} = 0.2125°$. Ker pa smo omejeni z natančnostjo rotacijske roke ($\varepsilon = 0.0514285\overline{5}°$), se lahko željeni vrednosti le približamo: $\theta_0 = n \cdot \varepsilon$, $n \in I\!N$. Eksperimentalni rezultati potrjujejo, da moramo v vseh izračunih uporabljati slednjo vrednost.

r predstavlja razdaljo med središčem sistema in optičnim centrom, ker pa lokacija optičnega centra navadno ni znana, smo razvili enostaven postopek, ki daje zadovoljive rezultate ocene parametra r. Eksperimentalni rezultati podajajo tudi možen postopek optimizacije.

Za oceno kota φ smo zapisali dva modela (enačba (A.2)): linearnega in nelinearnega. Za uporabljene kamere se je izkazalo, da sta oba modela približno enako dobra, ker pa je linearni model dal malenkost boljše rezultate, smo v drugih eksperimentih uporabljali slednji model.

Omejitev iskanja na epipolarni premici

Glede na to, da je širina panoramske slike veliko večja od širine standardne slike, kar pomeni, da je iskanje korespondenčne točke potrebno izvršiti vzdolž zelo dolge epipolarne premice, želimo ta preiskovalni prostor čimbolj omejiti. Tako povečamo zaupanje v ocenjeno globino in pospešimo delovanje sistema.

Če izhajamo iz enačbe (A.1), lahko zapišemo dve ugotovitvi, ki nam preiskovalni prostor močno omejita: 1) Če je θ_0 kot, za katerega premikamo kamero, je $2\theta_{\text{min}} = \theta_0$. Da pride točka, ki se nahaja v desni koloni in prispeva k panorami za levo oko, v levo kolono, moramo torej narediti obrat za vsaj en osnovni premik. 2) Teoretično velja, da ocena globine navzgor ni omejena, vendar iz enačbe (A.1) sledi, da imenovalec ne sme biti enak 0. To dejstvo lahko zapišemo kot: $\theta_{\text{maks}} = n \cdot \frac{\theta_0}{2}$, pri $n = \varphi \text{ div } \frac{\theta_0}{2}$ in $\varphi \bmod \frac{\theta_0}{2} \neq 0$. V nadaljevanju bomo pokazali, da v dobljeno oceno sicer ne moremo zaupati, vendar pa smo tukaj pokazali, da lahko tudi navzgor učinkovito omejimo preiskovalni prostor.

Tako na primer za naš sistem velja, da od 1501 slikovnih elementov, kolikor znaša širina panorame, ki se nahajajo na epipolarni premici, pregledamo le $n = 145$ slikovnih elementov pri $2\varphi = 29,9625°$ in le $n = 17$ slikovnih elementov pri $2\varphi = 3,6125°$.

Iz povedanega lahko sklepamo, da je postopek rekonstrukcije hitrejši, če je kot φ manjši, vendar pa bomo v nadaljevanju videli, da ima manjši kot φ tudi negativno lastnost.

Pomen napake za en slikovni element pri oceni kota θ

Ker je abscisna os grafa funkcije l v odvisnosti od θ diskretizirana na intervale velikosti $\frac{\theta_0}{2}$, je napaka za en slikovni element pri oceni kota θ (enačba (A.3)) veliko manjša pri večjem kotu φ.

Določitev največje globine, v katere oceno še zaupamo

	$2\varphi = 29,9625°$	$2\varphi = 3,6125°$
Δl_{min}	2 mm	19 mm
Δl_{maks}	30172 mm	81587 mm

Table A.1: Ilustracija napake (Δl) za en slikovni element pri oceni kota θ za oceno najmanjše možne globine l_{min} in največje možne globine l_{maks} glede na kot φ.

Tabela A.1 ilustrira napako za en slikovni element pri oceni kota θ za najmanjšo možno globino l_{min} in največjo možno globino l_{maks}. Vidimo lahko, da je vrednost napake Δl_{maks} nesprejemljivo visoka in to neglede na kot φ. Torej moramo smiselno zmanjšati oceno za največjo možno globino l_{maks}.

Doprinos vertikalne rekonstrukcije

Doprinos vertikalne rekonstrukcije na oceno razdalje l (enačbi (A.1) in (A.4)) je pozitiven, a majhen. V primeru uporabe kamere s horizontalnim zornim kotom $\alpha = 34°$, vertikalnim zornim kotom $\beta = 25°$ in parametri $r = 30$ cm, $2\varphi = 29.9625°$ in $\theta_0 = 0.205714°$, je ta 3,9 cm ali 1,8% od $l(\alpha, \beta)$. Pri tem smo definirali največjo dovoljeno napako Δl tako, da ta ne sme presegati 10 cm.

Vpliv različnih kamer

Vsako kamero lahko opišemo s horizontalnim zornim kotom α. Z večanjem horizontalnega zornega kota se širina panoramske slike manjša, višina pa ostaja enaka. Kotna ločljivost se torej manjša.

A.2.7 Eksperimentalni rezultati

V eksperimentih so uporabljene tri kamere: *kamera #1* z zornima kotoma $\alpha = 34°$ in $\beta = 25°$, *kamera #2* z zornima kotoma $\alpha = 39,72°$ in $\beta = 30,54°$ in *kamera #3* z zornima kotoma $\alpha = 16,53°$ in $\beta = 12,55°$. Korespondenčne točke za vsako izbrano značilko na sceni, ki jo uporabimo za oceno sistema, so bile določene s postopkom normalizirane korelacije (ang. *normalized correlation*) [4] ter preverjene ročno zaradi konsistentnosti. Rezultati eksperimentov so zbrani v tabeli A.2. Oceno, ki daje povprečno napako ocene globine l (enačbi (A.1) in (A.4)) v primerjavi z dejansko razdaljo d nad n izbranimi značilkami na sceni ter standardno odstopanje od te povprečne napake, izrazimo kot:

$$POVP_{\%} \pm SO_{\%} = \frac{\sum_{i=1}^{n} |l_i - d_i|/d_i}{n} \cdot 100\% \pm \sqrt{\frac{\sum_{i=1}^{n} \left(\frac{|l_i - d_i|}{d_i} \cdot 100\% - POVP_{\%} \right)^2}{n-1}}.$$

A.2.8 Zaključek

Na podlagi dosežene rekonstrukcijske točnosti lahko zaključimo, da osnovni sistem lahko uporabimo za samodejno lokalizacijo in navigacijo mobilnega robota v prostoru.

A.3 Delovanje v realnem času

A.3.1 Uvod

V splošnem lahko mozaične postopke gradnje večperspektivnih panoram označimo kot nesrediščne, ki se ne izvedejo v realnem času, a dajejo visoko ločljive rezultate z

1. Vpliv različnih vrednosti φ na rekonstrukcijsko točnost: *kamera #1*, računanje z $l(\alpha)$ (enačba (A.1)), $2\varphi = 3,6125°/2\varphi = 29,9625°$
11% ± 7,7%/**5% ± 2,6%**
2. Doprinos vertikalne rekonstrukcije (enačbi (A.1) in (A.4)): *kamera #1*, $l(\alpha)/l(\alpha, \beta)$
5% ± 2,6%/**4,7% ± 2,7%**
3. Vpliv različnih vrednosti θ_0 na rekonstrukcijsko točnost: *kamera #1*, od tu naprej računamo z $l(\alpha, \beta)$ (enačba (A.4)), $\theta_0 = 0,205714°/\theta_0 = 0,2125°$
4,7% ± 2,7%/22,8% ± 19%
4. Primerjava linearnega in nelinearnega modela za oceno kota φ (enačba (A.2)): *kamera #1*, $2\varphi = 29,9625°/2\varphi = 30,15774565°$
4,7% ± 2,7%/5,3% ± 3,5%
5. Ponovljivost rezultatov – drug prostor: *kamera #1*, primerjaj z 2. in 6. eksperimentom
4,5% ± 4,5%
6. Ponovljivost rezultatov – različne kamere, tretji prostor: *kamera #1*/*kamera #2*/*kamera #3*
3% ± 2%/2,7% ± 2,3%/5,9% ± 2,5%
7. Možnost prisotnosti sistematske napake pri oceni parametra r: *kamera #3*, brez optimizacije/po optimizaciji (minimizacija $POVP_{\%}$)
5,9% ± 2,5%/**2,2% ± 1,5%**
8. Vpliv prisotnosti distorzije leč na rekonstrukcijsko točnost: *kamera #2*, pred odpravo popačenja/po odpravi popačenja
2,7% ± 2,3%/**2,4% ± 1,6%**

Table A.2: Rezultati eksperimentov: kratkemu opisu eksperimenta sledijo vrednosti najpomembnejših parametrov ter ocena $POVP_{\%} \pm SO_{\%}$. Podrobneje so ozadja eksperimentov razložena v predhodnih poglavjih.

enako ločljivostjo v vseh smereh panoramske slike. Torej ti postopki niso primerni za zajemanje dinamičnih scen.

Panoramski globinski senzor, ki deluje na osnovi večperspektivnih panoramskih slik in deluje v realnem času, v literaturi še ni bil predstavljen in ocenjen s stališča učinkovitosti. V tem poglavju bomo predlagali nov (nadgrajen) senzor, ki omogoča delovanje v realnem času in ocenili njegovo učinkovitost.

Figure A.2: Širši je trak, več točk na sceni je takih, ki niso zajete v panoramski sliki: leva panoramska slika je bila zgrajena iz le ene kolone vsake zajete slike, desna panoramska slika pa je bila zgrajena iz trakov širine 14 kolon. Opazimo lahko, da so vse oddaljene točke na obeh slikah dobro zajete, medtem ko nekatere točke (tekst na škatli) niso zajete na desni sliki.

A.3.2 Gradnja panoramskih slik iz širših trakov

Prva ideja, kako zgraditi stereo par hitreje, je gradnja iz širših trakov in ne le ene kolone zajete slike. Ta postopek je seveda veliko hitrejši, vendar zahteva od nas sprejem kompromisa med hitrostjo gradnje para in njegovo kvaliteto.

Lastnost uporabe trakov

Če opazujemo panoramsko sliko zgrajeno iz trakov bolj podrobno, lahko opazimo, da slika ni popolna. Pri tem ne mislimo na možne šive med trakovi ali popačenja zaradi lastnosti leče na kameri. Bistven problem lahko opazimo na bližnjih objektih, ki imajo lepo teksturo (na primer tekst). V takšnih primerih lahko vidimo, da določene točke scene niso zajete A.2.

Če vzamemo pod drobnogled dogajanje med dvema zaporednima korakoma sistema, vidimo, da trakovi, ki doprinesejo k panoramski sliki, ne pokrivajo celotne scene. To velja tudi za panoramske slike, zgrajene iz ene kolone vsake zajete slike. Vendar, če želimo zajeti paralakso gibanja, moramo sprejeti tudi dejstvo, da vseh točk scene ne moremo zajeti.

A.3.3 Doseganje realnega časa

V osnovnem sistemu zgradimo panoramsko sliko na osnovi premikanja standardne kamere za zelo majhen kot po vnaprej definirani krožni poti. Če bi imeli kamero na vsaki lokaciji te poti, bi lahko panoramsko sliko zgradili v realnem času. Žal pa v praksi ne moremo dati toliko standardnih kamer tako blizu skupaj (glede na razumno velikost radija r). V primeru, da bi gradili panoramsko sliko iz slik ločljivosti 160×120 slikovnih elementov, potem bi morali kamere s horizontalnim zornim kotom $\alpha = 34°$ razmakniti med seboj za kot 0.205714°. Torej bi za zajetje celotne panoramske slike potrebovali 1750 kamer.

Če uporabimo širše trakove, potem število potrebnih kamer postane sprejemljivejše. Tako trak širine 14 kolon prinese razmik med kamerami enak 2.879999° in

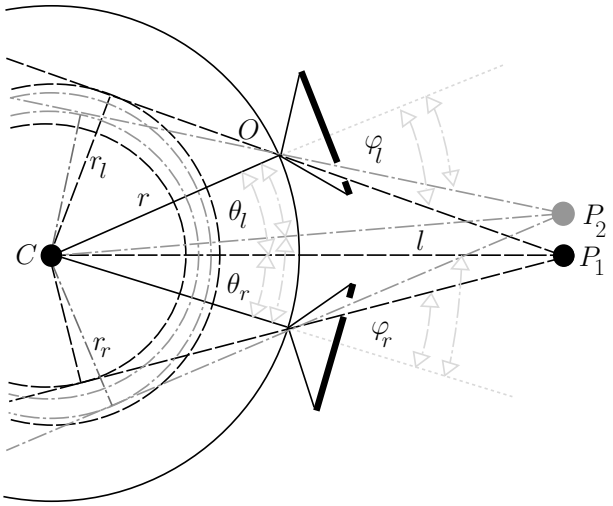


Figure A.3: Kota θ_l in θ_d sta povezana s kotoma φ_l in φ_d kot to podaja enačba (A.6). Relacija je tukaj ilustrirana za dve točki na sceni.

potrebo bo 125 kamerah, ki pokrijejo celotno krožno pot. V primeru uporabe kamer s širšim zornim kotom (npr. $\alpha = 90°$) potrebujemo ustrezno manj kamer (npr. 46). Nov senzor tako ne potrebuje pomičnih delov, kar pomeni, da nimamo problemov z vibracijami sistema in da nismo omejeni z radijem krožne poti. Slednja ugotovitev nam omogoča, da zgradimo sistem iz standarnih kamer, ki so dosegljive na tržišču.

A.3.4 Stereo rekonstrukcija iz trakov

Enačba (A.1) je zapisana za primer gradnje simetričnega para stereo panoramskih slik iz le ene kolone vsake zajete slike. Toda ob gradnji iz trakov moramo enačbe prilagoditi novemu načinu gradnje. V tem primeru gradimo iz simetričnih trakov vsake zajete slike. Medtem ko smo kolono definirali s kotom φ, je trak definiran z dvema takšnima kotoma: φ_{min} in φ_{maks}. V panorami za levo oko lahko tako vsakemu slikovnemu elementu znotraj traku pripišemo kot φ_l: $\varphi_{\text{min}} \leq \varphi_l \leq \varphi_{\text{maks}}$. Ko najdemo korespondenčno točko na panorami za desno oko, lahko ocenimo kot φ_d na enak način, glede na lokacijo korespondenčne točke znotraj traku: $\varphi_{\text{min}} \leq \varphi_d \leq \varphi_{\text{maks}}$.

Po analogiji z enačbo (A.3) in upoštevanjem dejstva, da gradimo panorame iz trakov, lahko zapišemo naslednjo enačbo (sliki A.1 in A.3):

$$2\theta = \theta_l + \theta_d. \tag{A.5}$$

Ob uporabi trakov širine ene kolone zajete slike, je $\theta_l = \theta_d$ (slika A.1), vendar temu ni nujno tako, če uporabljamo širše trakove. V splošnem sta ti dve vrednosti različni,

vendar lastnost, ki sledi iz enačbe

$$\frac{\theta_l}{\theta_d} = \frac{\varphi_l}{\varphi_d}, \tag{A.6}$$

nakazuje, da je razmerje teh dveh vrednosti povezano z vrednostima kotov φ (slika A.3). Večji kot je kot φ_l, večji je tudi kot θ_d. Sedaj lahko enostavno izrazimo θ_d in θ_l iz enačb (A.3), (A.5) in (A.6) kot:

$$\theta_d = \frac{dx \cdot \theta_0}{(1+\frac{\varphi_l}{\varphi_d})}$$
$$\theta_l = dx \cdot \theta_0 - \theta_d.$$

Ker vemo, da večji φ prinaša večjo rekonstrukcijsko točnost (poglavje A.2.7), vedno pri računanju globine l (enačbi (A.1) in (A.4)) uporabimo večji φ od dveh možnih (φ_l in φ_d).

Eksperimentalni rezultati v poglavju A.3.6 potrjujejo pravilnost principa.

A.3.5 Epipolarna omejitev

Trak je sestavljen iz kolon, kjer je vsaka definirana s svojim kotom φ. V osnovi to pomeni, da imamo opravka z nesimetričnimi pari panoram, za katere epipolarne premice niso vrste v panoramski sliki. Vendar lahko na zadevo pogledamo tudi z drugega zornega kota: Panorame gradimo iz simetričnih trakov. Če zmanjšamo ločljivost zajetih slik, se trak spremeni v kolono. Simetrični trakovi tako postanejo simetrične kolone in spet lahko uporabimo vrste panoramske slike za epipolarne premice.

Obstaja pa tudi možnost rekonstrukcije scene iz nesimetričnih panoram, kjer lahko še vedno uporabljamo enostavno horizontalno epipolarno geometrijo [20] (poglavje A.3.6).

A.3.6 Eksperimentalni rezultati

Osnove o tem, kako so rezultati dobljeni, ocenjeni ipd., so podane v poglavju A.2.7. Delovanje v realnem času je bilo simulirano tako, da smo vrteli eno standardno kamero za kot, ki ustreza širini traku W_t. Rezultati eksperimentov so zbrani v tabeli A.3. Ugotovili smo tudi, da odprava distorzije leč v primeru gradnje iz trakov ne prinese izboljšanja rezultatov.

A.3.7 Zaključek

Narejene predpostavke so se izkazale za pravilne in razkrile nekaj zanimivih lastnosti novega sistema. Žal pa smo omejeni pri samem apliciranju: odvisnost med povprečno napako $POVP_{\%}$ in širino traku W_t ($POVP_{\%}(W_t)$) je približno linearna in

1. Rekonstrukcija na osnovi nesimetričnega para panoram: *kamera #1*, povsod računamo z $l(\alpha, \beta)$ (enačba (A.4)), simetričen par/nesimetričen par ($2\varphi_l = 29.9625°$, $2\varphi_d = 3.6125°$)
3% ± 2%/3,3% ± 1,9%
2. Rekonstrukcija iz panoram dobljenih na osnovi trakov: *kamera #1*, $W_t = 2/W_t = 6/W_t = 14/W_t = 29$
2,9% ± 2,7%/4,5% ± 4%/11,9% ± 7,9%/22,1% ± 15,8%
3. Ponovljivost rezultatov – različne kamere: $W_t = 14$, *kamera #1/kamera #2/kamera #3*
11,9% ± 7,9%/19,4% ± 15,3%/10% ± 6,6%
4. Ponovljivost rezultatov – drug prostor: *kamera #1*, $W_t = 6/W_t = 14$
5,4% ± 4,5%/16% ± 19,6%

Table A.3: Rezultati eksperimentov: kratkemu opisu eksperimenta sledijo vrednosti najpomembnejših parametrov ter ocena $POVP_{\%} \pm SO_{\%}$. Podrobneje so ozadja eksperimentov razložena v predhodnih poglavjih.

naraščujoča, pri konstantni širini traku W_t pa se z večanjem horizontalnega zornega kota α slabša rekonstrukcijska točnost $POVP_{\%}$.

A.4 Sklep

Glavni cilj disertacije je odgovoriti na vprašanje ali so rezultati našega sistema uporabni za samodejno lokalizacijo in navigacijo mobilnega robota v prostoru. Do odgovora smo prišli na podalagi analize sistema: Glede na rekonstrukcijsko točnost osnovnega sistema lahko zaključimo, da je sistem vsekakor možno uporabiti za takšne naloge. Ker je v osnovnem sistemu zajemanje stereo para panoramskih slik dolgotrajen postopek, smo sistem ustrezno nadgradili v tej smeri. Predpostavke narejene ob nadgradnji so se izkazale za pravilne, vendar pa se rekonstrukcijska točnost novega sistema navadno zmanjša v primerjavi z osnovnim sistemom.

A.4.1 Prispevki k znanosti

Bistveni prispevki, ki niso bili objavljeni s strani drugih avtorjev [38, 47, 58], so:

- Pokazali smo, da lahko učinkovito omejimo preiskovalni prostor na epipolarni premici (poglavje A.2.6).
- Pokazali smo, da je zaupanje v ocenjeno globino spremenljivo glede na kot φ (poglavji A.2.6 in A.2.7).

- Pokazali smo, da horizontralni zorni kot α kamere podaja specifično število ocen globin na kotno stopinjo (poglavje A.2.6).
- Predlagali in ocenili smo učinkovitost sistema, ki deluje v realnem času (poglavje A.3).
- Pokazali smo, da tako kolone kot trakovi, iz katerih gradimo panoramske slike, ne zajamejo celotne scene (poglavje A.3.2).
- Pokazali smo, da se rekonstrukcijska točnost zmanjšuje približno linearno z večanjem širine traku. Velja tudi, da se pri konstantni širini traku W_t (> 1) in večanjem horizontalnega zornega kota α rekonstrukcijska natančnost manjša (poglavje A.3.6).
- Preučili smo vpliv distorzije leč na rekonstrukcijsko točnost (poglavji A.2.7 in A.3.6).
- In nenazadnje, disertacija lahko služi kot vodič za gradnjo panoramskih globinskih senzorjev in sorodnih sistemov.

Bibliography

1992

[1] Ishiguro H., Yamamoto M., Tsuji S.: Omni-directional stereo. *IEEE Trans. PAMI*, 14(2), 257–262.

[2] Paar G., Pölzleitner W.: Robust disparity estimation in terrain modeling for spacecraft navigation. Proc. *IEEE ICPR*, The Hague, The Netherlands, August 30 – September 3, I:738–741.

[3] Weng J., Cohen P., Herniou M.: Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. PAMI*, 14(10), 965–980.

1993

[4] Faugeras O.: *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts, London, England.

[5] Faugeras O., Hotz B., Mathieu H., Viéville T., Zhang Z., Fua P., Théron E., Moll L., Berry G., Vuillemin J., Bertin P., Proy C.: Real time correlation based stereo: algorithm implementations and applications. *Technical Report 2013*, INRIA Sophia-Antipolis, France. Available at: ftp://ftp-sop.inria.fr/pub/rapports/RR-2013.ps.gz .

[6] Okutomi M., Kanade T.: A Multiple-Baseline Stereo. *IEEE Trans. PAMI*, 15(4), 353–363.

1995

[7] Basu A., Licardie S.: Alternative models for fish-eye lenses. *Pattern Recognition Letters*, 16(4), 433–441.

[8] Chen S.: Quicktime VR — an image-based approach to virtual environment navigation. Proc. *ACM SIGGRAPH*, Los Angeles, USA, August 6–11, 29–38.

1996

[9] Szeliski R.: Video Mosaics for Virtual Environments, *IEEE Computer Graphics and Applications*, 16(2), 22–30.

1997

[10] Gupta R., Hartley R. I.: Linear pushbroom cameras. *IEEE Trans. PAMI*, 19(9), 963–975.

[11] Heikkilä J., Silvén O.: A Four-Step Camera Calibration Procedure with Implicit Image Correction. Proc. *IEEE CVPR*, San Juan, Puerto Rico, June 17–19, 1106–1112.

[12] Szeliski R., Shum H. Y.: Creating full view panoramic image mosaics and texture-mapped models. *Computer Graphics (ACM SIGGRAPH)*, Los Angeles, USA, August 3–8, 251–258.

[13] Wood D., Finkelstein A., Hughes J., Thayer C., Salesin D.: Multiperspective panoramas for cel animation. *Computer Graphics (ACM SIGGRAPH)*, Los Angeles, USA, August 3–8, 243–250.

1998

[14] Benosman R., Maniere T., Devars J.: Panoramic stereovision sensor. Proc. *IEEE ICPR*, Brisbane, Australia, August 16–20, I:767–769.

[15] Gluckman J., Nayar S. K., Thorek K. J.: Real-time omnidirectional and panoramic stereo. Proc. *DARPA Image Understanding Workshop*, Monterey, USA, November.

[16] Prihavec B., Solina F.: User interface for video observation over the internet. *Journal of Network and Computer Applications*, 21, 219–237.

[17] Rademacher P., Bishop G.: Multiple-center-of-projection images. *Computer Graphics (ACM SIGGRAPH)*, Orlando, USA, July 19–24, 199–206.

1999

[18] Nayar S. K., Peri V.: Folded Catadioptric Camera. Proc. *IEEE CVPR*, Fort Collins, USA, June 23–25, II:217–223.

[19] Peleg S., Ben-Ezra M.: Stereo panorama with a single camera. Proc. *IEEE CVPR*, Fort Collins, USA, June 23–25, I:395–401.

[20] Shum H. Y., Szeliski R.: Stereo Reconstruction from Multiperspective Panoramas. Proc. *IEEE ICCV*, Kerkyra, Greece, September 20–25, I:14–21.

[21] Shum H. Y., Kalai A., Seitz S. M.: Omnivergent Stereo. Proc. *IEEE ICCV*, Kerkyra, Greece, September 20–25, I:22–29.

2000

[22] Hartley R., Zisserman A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK.

[23] Huang F., Pajdla T.: Epipolar geometry in concentric panoramas. *Technical Report CTU-CMP-2000-07*, Center for Machine Perception, Czech Technical University, Prague, Czech Republic. Available at: ftp://cmp.felk.cvut.cz/pub/cmp/articles/pajdla/Huang-TR-2000-07.ps.gz .

[24] Jogan M., Leonardis A.: Robust localization using the eigenspace of spinning-images. Proc. *IEEE Workshop on Omnidirectional Vision*, Hilton Head Island, USA, June 12, 37–44.

[25] Nayar S. K., Karmarkar A.: 360×360 Mosaics. Proc. *IEEE CVPR*, Hilton Head Island, USA, June 13–15, II:388–395.

[26] Peleg S., Pritch Y., Ben-Ezra M.: Cameras for stereo panoramic imaging. Proc. *IEEE CVPR*, Hilton Head Island, USA, June 13–15, I:208–214.

[27] Peleg S., Rousso B., Rav-Acha A., Zomet A.: Mosaicing on adaptive manifolds. *IEEE Trans. PAMI*, 22(10), 1144–1154.

[28] Svoboda T.: Central Panoramic Camera Design, Geometry, Egomotion. *Ph.D. Thesis*, Center for Machine Perception, Czech Technical University, Prague, Czech Republic. Available at: ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/phdthesis.ps.gz .

[29] Svoboda T., Pajdla T.: Panoramic cameras for 3D computation. Proc. *Czech Pattern Recognition Workshop*, Prague, Czech Republic, 63–70.

[30] Tanahashi H., Yamamoto K., Wang C., Niwa Y.: Development of a Stereo Omnidirectional Imaging System (SOS). Proc. *IEEE International Conference on Industrial Electronics, Control and Instrumentation*, Nagoya, Japan, October 22–28, 289–294.

[31] Zhang Z.: A flexible new technique for camera calibration. *IEEE Trans. PAMI*, 22(11), 1330-1334.

2001

[32] Bakstein H., Pajdla T.: 3D Reconstruction from 360×360 Mosaics. Proc. *IEEE CVPR*, Kauai, Hawaii, USA, December 8–14, I:72–77.

[33] Benosman R., Kang S. B. (Eds.): *Panoramic Vision: Sensors, Theory and Applications*. Springer-Verlag, New York, USA.

[34] Faugeras O., Luong Q.-T.: *The Geometry of Multiple Images*. MIT Press, Cambridge, Massachusetts, London, England.

[35] Huang F., Wei S. K., Klette R.: Geometrical Fundamentals of Polycentric Panoramas. Proc. *IEEE ICCV*, Vancouver, Canada, July 9–12, I:560–565.

[36] Li Y., Tang C. K., Shum H. Y.: Efficient Dense Depth Estimation from Dense Multiperspective Panoramas. Proc. *IEEE ICCV*, Vancouver, Canada, July 9–12, I:119–126.

[37] Pajdla T.: Epipolar geometry of some non-classical cameras. Proc. *Computer Vision Winter Workshop (CVWW)*, Bled, Slovenia, February 7–9, 223–233.

[38] Peer P., Solina F.: Capturing mosaic-based panoramic depth images with a single standard camera. *International Journal of Machine Graphics and Vision*, 10(3), 369–397.

[39] Peleg S., Ben-Ezra M., Pritch Y.: Omnistereo: Panoramic Stereo Imaging. *IEEE Trans. PAMI*, 23(3), 279–290.

[40] Seitz S. M.: The Space of All Stereo Images. Proc. *IEEE ICCV*, Vancouver, Canada, July 9–12, I:26–33.

[41] Shimada D., Tanahashi H., Kato K., Yamamoto K.: Extract and Display Moving Object in All Direction by Using Stereo Omnidirectional System (SOS). Proc. *IEEE International Conference on 3-D Digital Imaging and Modeling*, Quebec City, Canada, May 28 – June 1, 42–47.

[42] Tanahashi H., Shimada D., Yamamoto K., Niwa Y.: Acquisition of Three-Dimensional Information in Real Environment By Using Stereo Omni-directional System (SOS). Proc. *IEEE International Conference on 3-D Digital Imaging and Modeling*, Quebec City, Canada, May 28 – June 1, 365–371.

[43] Wei S. K., Huang F., Klette R.: Determination of geometric parameters for stereoscopic panorama cameras. *International Journal of Machine Graphics and Vision*, 10(3), 399–427.

2002

[44] Bakstein H., Pajdla T.: Panoramic Mosaicing with a 180° Field of View Lens. Proc. *IEEE Workshop on Omnidirectional Vision*, Copenhagen, Denmark, June, 60–67.

[45] Hirschmüller H., Innocent P. R., Garibaldi J.: Real-Time Correlation-Based Stereo Vision with Reduced Border Errors. *International Journal of Computer Vision*, 47(1/2/3), 229–246.

[46] Pajdla T.: Stereo with Oblique Cameras. *International Journal of Computer Vision*, 47(1/2/3), 161–170.

[47] Peer P., Solina F.: Panoramic Depth Imaging: Single Standard Camera Approach. *International Journal of Computer Vision*, 47(1/2/3), 149–160.

[48] Scharstein D., Szeliski R.: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1/2/3), 7–42.

[49] Seitz S. M., Kim J.: The Space of All Stereo Images. *International Journal of Computer Vision*, 48(1), 21–38.

[50] Shah M.: Guest Introduction: The Changing Shape of Computer Vision in the Twenty-First Century. *International Journal of Computer Vision*, 50(2), 103–110.

[51] Sivic J.: Geometry of Concentric Multiperspective Panoramas. *M.Sc. Thesis*, Center for Machine Perception, Czech Technical University, Prague, Czech Republic.

[52] Svoboda T., Pajdla T.: Epipolar Geometry for Central Catadioptric Cameras. *International Journal of Computer Vision*, 49(1), 23–37.

2003

[53] Bakstein H., Pajdla T.: Ray space volume of omnidirectional 180°×360° images. Proc. *Computer Vision Winter Workshop (CVWW)*, Valtice, Czech Republic, February 3–6, 39–44.

[54] Bouguet J.-Y.: Camera Calibration Toolbox for Matlab. California Institute of Technology. Available at: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html .

[55] Brown M. Z., Burschka D., Hager G. D.: Advances in Computational Stereo. *IEEE Trans. PAMI*, 25(8), 993–1008.

[56] Feldman D., Zomet A., Weinshall D., Peleg S.: New view synthesis with non-stationary mosaicing. Proc. *Computer Vision / Computer Graphics Collaboration for Model-based Imaging, Rendering, image Analysis and Graphical special Effects (MIRAGE)*, INRIA Rocquencourt, France, March 10–11, 48–56.

[57] Matsuyama T., Wu X., Takai T., Nobuhara S.: Real-Time Generation and High Fidelity Visualization of 3D Video. Proc. *Computer Vision / Computer Graphics Collaboration for Model-based Imaging, Rendering, image Analysis and Graphical special Effects (MIRAGE)*, INRIA Rocquencourt, France, March 10–11, 1–10.

[58] Peer P., Solina F.: Towards a Real Time Panoramic Depth Sensor. Proc. *International Conference on Computer Analysis of Images and Patterns (CAIP)*, Groningen, The Netherlands, August 25–27, 107–115.

[59] Skočaj D.: Robust subspace approaches to visual learning and recognition. *Ph.D. Thesis*, University of Ljubljana, Faculty of Computer and Information Science. Available at: http://eprints.fri.uni-lj.si .

[60] Sun C., Peleg S.: Fast Panoramic Stereo Matching Using Cylindrical Maximum Surfaces. *IEEE Trans. on Systems, Man and Cybernetics – Part B*, Accepted for publication.

[61] Zomet A., Feldman D., Peleg S., Weinshall D.: Mosaicing New Views: The Crossed-Slits Projection. *IEEE Trans. PAMI*, 25(6), 741–754.

Acknowledgement

Prof. dr. Franc Solina stood beside me from the beginning of my study in 1993, when I enroled at the Faculty of Computer and Information Science, University of Ljubljana. Under his supervision I have gained a B.Sc. and a M.Sc. degree. He is the one that introduced the field of computer vision to me and without him this disseration definitely would not be written. He is a supervisor one could only wish and I am very grateful and proud that I am one of his students.

Prof. Solina formed the Computer vision laboratory in 1991. When I came to the laboratory in 1993 there were only a few people working there. Nowadays, the number of people working in the laboratory is much bigger and the laboratory is well recognized in the computer vision community. Here I would like to express my gratitute to all present and former members of the laboratory. They are always available when I need them.

I would also like to thank prof. dr. Václav Hlaváč, prof. dr. Stanislav Kovačič and prof. dr. Aleš Leonardis for serving as the members of the dissertation committee.

Statement

I hereby state that this dissertation has been elaborated independently under the supervision of prof. dr. Franc Solina. All the help from others is alleged in the acknowledgement.

Ljubljana, 8 December 2003 Peter Peer