# Chapter 1

# Multiperspective panoramic depth imaging

**Peter Peer, Franc Solina**

*Computer Vision Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, 1001 Ljubljana, Slovenia*

e-mail: {peter.peer, franc.solina}@fri.uni-lj.si

Abstract

In this chapter we present a stereo panoramic depth imaging system, which builds depth panoramas from multiperspective panoramas while using only one standard camera.

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 system has been comprehensively analysed. The analyses include the study of influence of different system parameters on the reconstruction accuracy, constraining the search space on the epipolar line, meaning of error in estimation of corresponding point, definition of the maximal reliable depth value, contribution of the vertical reconstruction and influence of using different cameras. They are substantiated with a number of experiments, including experiments addressing the baseline, the repeatability of results in different rooms, by using different cameras, influence of lens distortion presence on the reconstruction accuracy and evaluation of different models for estimation of system parameters. The analyses and the experiments revealed a number of interesting properties of the system.

According to the basic system accuracy we definitely can use the system for autonomous robot localization and navigation tasks.

**Keywords:** Computer vision, Stereo vision, Reconstruction, Depth image, Multiperspective panoramic image, Mosaicing, Motion parallax effect, Standard camera, Depth sensor

## 1.1 Introduction

### 1.1.1 Motivation

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 chapter we present a stereo panoramic depth imaging system.

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].

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.

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. We do the stereo reconstruction from two symmetric multiperspective panoramic images.

In this chapter 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. Accordingly, in this chapter we present 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.

### 1.1.2 Basics about the system

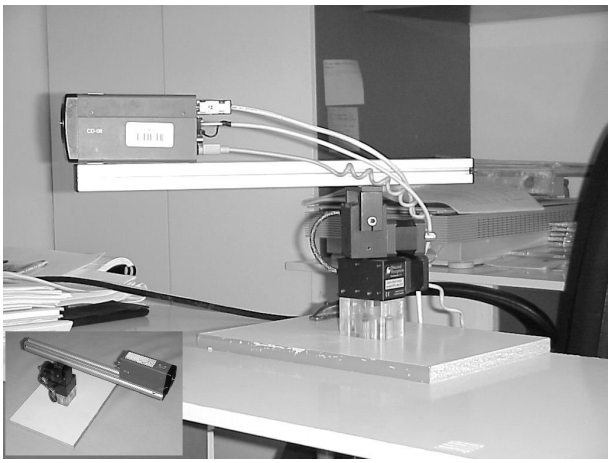


Figure 1.1: Hardware part of our system.

In Fig. 1.1 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 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.

### 1.1.3 Structure of the chapter

In the next section we compare different panoramic cameras with emphasis on mosaicing. In Sec. 1.3 we give an overview of related work and briefly present the contribution of our work towards the discussed subject. Sec. 1.4 describes the geometry of our system, Sec. 1.5 is devoted to the epipolar geometry and Sec. 1.6 describes the procedure of stereo

reconstruction. The focus of this chapter is on the analysis of system capabilities, given in Sec. 1.7. In Sec. 1.8 we present experimental results. In the very end of this chapter we summarize the main conclusions.

## 1.2 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 1.1: 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. 1.1.

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.

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.

## 1.3 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.

## 1.4 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 chapter, 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].

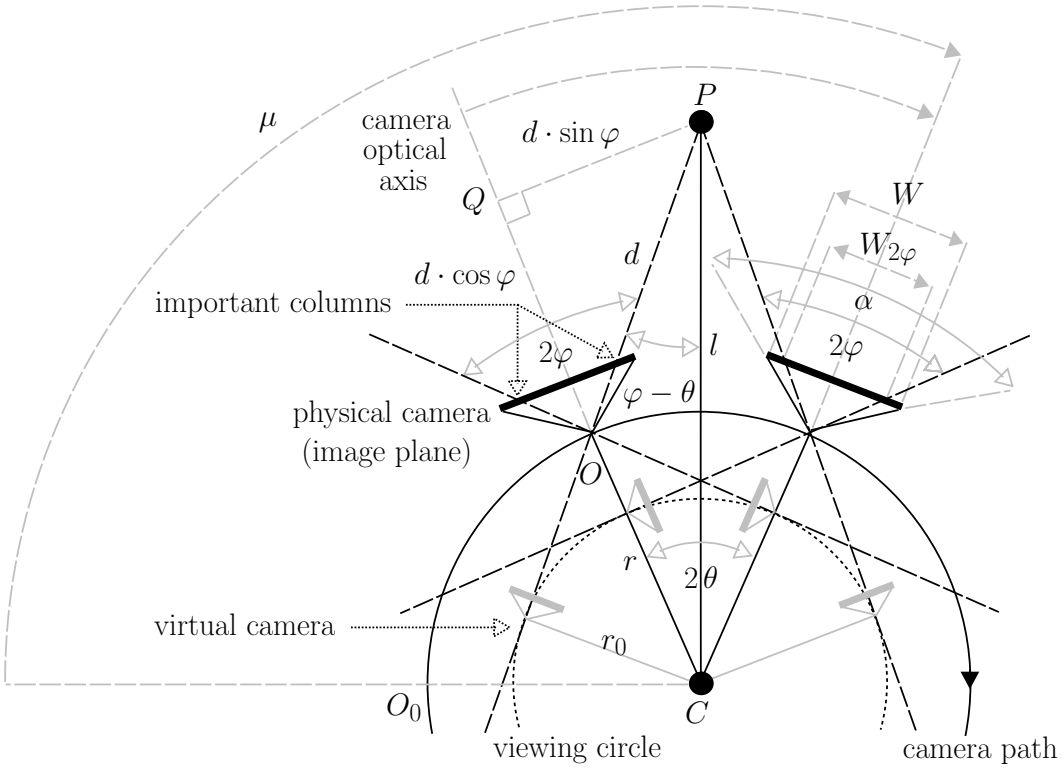



Figure 1.2: Geometry of our system for constructing multiperspective panoramic images. Note that a ground-plan is presented. The optical axis of the camera is kept horizontal.

The geometry of our system for creating multiperspective panoramic images is shown in Fig. 1.2. 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, $\theta$ 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 $\mu$ and defined by point $C$, a starting point $O_0$, where we start capturing the panoramic image, and the current point $O$. $\varphi$ 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 $\varphi$ can be thought of as a reduction of the camera's horizontal view angle $\alpha$.

The geometry of capturing multiperspective panoramic images can be described with a pair of parameters $(r, \varphi)$. By increasing (decreasing) each of them, we increase (decrease) the baseline ($2r_0$ [39], $r_0 = r \cdot \sin\varphi$ (Fig. 1.2)) of our stereo system.

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

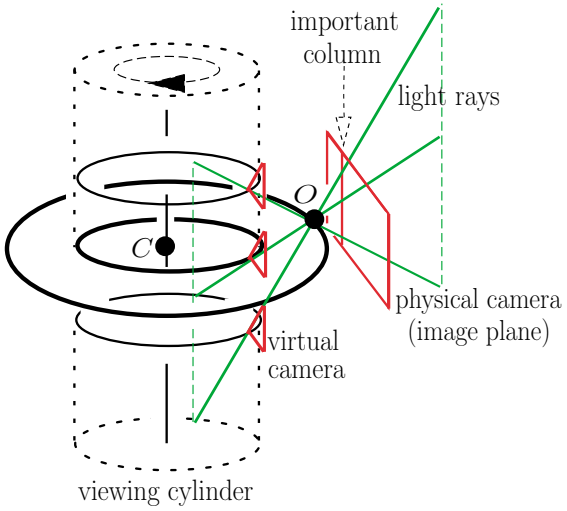
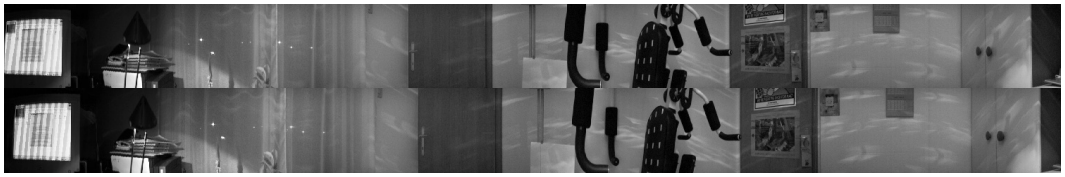



Figure 1.3: All the light rays forming the panoramic image are tangent to the viewing cylinder.

$2\varphi = 29.9625°$



$2\varphi = 3.6125°$

Figure 1.4: Two symmetric pairs of panoramic images generated using different values of the angle $\varphi$. In Sec. 1.7.1 we explain where these values for the angle $\varphi$ come from. Each symmetric pair of panoramic images comprises the motion parallax effect. This fact enables the stereo reconstruction.

The system in Fig. 1.2 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. 1.3). 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. 1.2) then we can conclude that our system is equivalent 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. 1.3).

Two images differing in the angle of rotation of the physical camera setup (for example, two image planes marked in Fig. 1.2) 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. 1.4 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 $\alpha$ and vertical view angle $\beta$ 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 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 $0.0514285\overline{5}\,^\circ$. 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:

$$\alpha = \frac{W}{dx} \cdot d\gamma, \tag{1.1}$$

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:

$$\alpha = 2 \arctan \frac{W/2}{f}, \tag{1.2}$$

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 $\alpha$, 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 $\theta_0$:

$$W_s = x_0 \cdot \theta_0. \tag{1.3}$$

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.

## 1.5 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. 1.7.3 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. 1.2 and 1.3). We can express $r_0$ in the terms of known parameters $r$ and $\varphi$ 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.

## 1.6 Stereo reconstruction

Let us go back to Fig. 1.2. 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:

$$\frac{r}{\sin(\varphi-\theta)} = \frac{d}{\sin\theta} = \frac{l}{\sin(180^\circ-\varphi)}. \tag{1.4}$$

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

$$l = \frac{r\cdot\sin(180^\circ-\varphi)}{\sin(\varphi-\theta)} = \frac{r\cdot\sin\varphi}{\sin(\varphi-\theta)}. \tag{1.5}$$

Eq. (1.5) implies that we can estimate depth $l$ only if we know three parameters: $r$, $\varphi$ and $\theta$. $r$ is given. Angle $\varphi$ can be calculated on the basis of the camera's horizontal view angle $\alpha$ (Eq. (1.1)) as:

$$2\varphi = \frac{\alpha}{W}\cdot W_{2\varphi}, \tag{1.6}$$

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 $\theta$, 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 $\theta_0$, we can express the angle $\theta$ as:

$$\theta = dx\cdot\frac{\theta_0}{2}, \tag{1.7}$$

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. (1.5) does not contain the focal length $f$ explicitly, but since the relationships between $\alpha$ and $f$ on one side (Eq. (1.2)) and $\alpha$ and $\varphi$ on the other side (Eq. (1.6)) exist, $\varphi$ also depends upon $f$ (the two models for estimating angle $\varphi$ (Eqs. (1.6) and (1.8)) are discussed in Sec. 1.7.2):

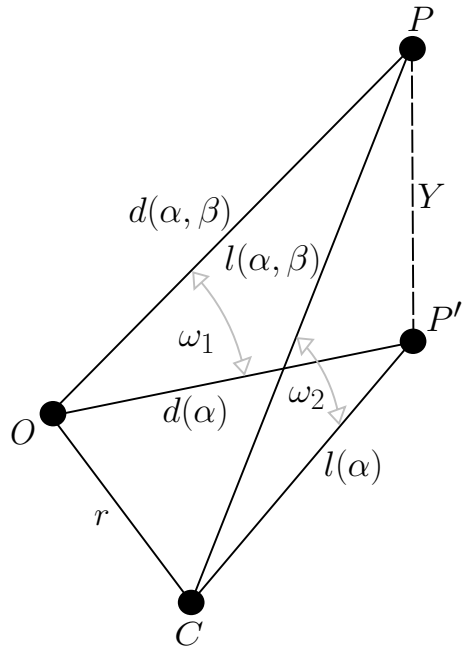



Figure 1.5: Important relations between system parameters for addressing the vertical reconstruction.

$$\varphi = \arctan \frac{W_{2\varphi}/2}{f}. \tag{1.8}$$

Eq. (1.5) 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 of the scene point $P$ is marked with $P'$ in Fig. 1.5. 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 $\beta$ into Eq. (1.5).

Let us here adopt the following notation to introduce the influence of $\beta$ on estimation of $l$: if a variable $l$ or $d$ depends on $\alpha$ 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 $\alpha$ and $\beta$, we mark that as $l(\alpha, \beta)$ and $d(\alpha, \beta)$. According to Fig. 1.5 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 $\omega_2$ 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 $\omega_1$ similarly as we calculated $\varphi$ (Eqs. (1.6) and (1.8)):

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

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. (1.4):

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

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

$$l(\alpha, \beta) = \sqrt{l(\alpha)^2 + \left(\frac{l(\alpha) \cdot \sin\theta}{\sin\varphi} \cdot \tan\omega_1\right)^2}. \tag{1.9}$$

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. 1.7.6 and 1.8.4.

## 1.7 Analysis of the system's capabilities

### 1.7.1 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 $160 \times 120$ pixels, where radius is $r = 30$ cm and the shift angle is $\theta_0 = 0.205714°$. We have choosen the resolution of $160 \times 120$ 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 $2\varphi$ (Eq. (1.6)) 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.

### 1.7.2 Influence of parameters $r$, $\varphi$ and $\theta_0$ 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 $r$, $\varphi$ and $\theta_0$.

$\theta_0$ denotes the angle corresponding to one pixel column of the captured image, for which we rotate the camera. It can be calculated from Eq. (1.3):

$$\theta_0 = \frac{\alpha}{W}. \tag{1.10}$$

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 $\varepsilon = 0.0514285\overline{7}°$, so the best possible approximate value is $\theta_0 = 0.205714°$. Since each column in the panoramic image in reality describes the latter angle $\theta_0$, we always use in calculations $\theta_0 = n \cdot \varepsilon$, $n \in IN$, which is closest to the result obtained from Eq. (1.10). The experiment in Sec. 1.8.5 confirms that this decision is correct. To discriminate the two values between each other, let us mark them as $\theta_0(\alpha)$ (Eq. (1.10)) and $\theta_0(\varepsilon)$ (the estimation based on the accuracy of our rotational arm). We use them from now on, but where only $\theta_0$ is given, then $\theta_0 = \theta_0(\varepsilon)$.

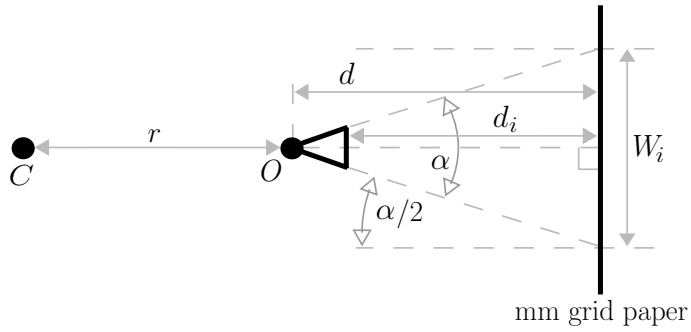



Figure 1.6: The relation between the parameters, which are important for determining the radius $r$.

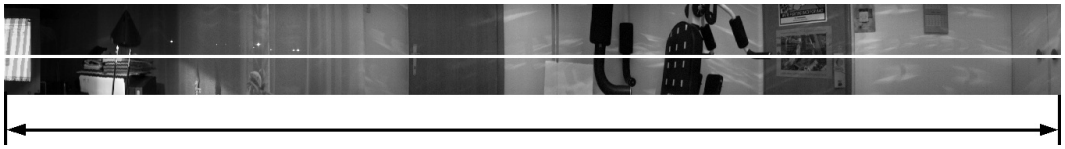
$r$ represents the distance between the rotational center of the system and the optical center of the camera. Since the exact position of the optical center is normally not known (not given by the manufacturer), we have to estimate its position. Optical firms with their special equipment would do the 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. 1.6): First the camera horizontal view angle $\alpha$ 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 ($\alpha$, $d_i$ 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. 1.8.9.

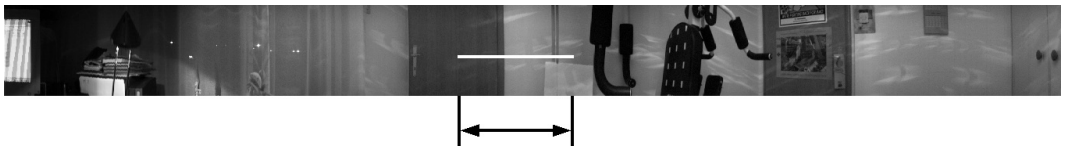
$\varphi$ determines the column of each captured image, which is mosaiced into the panoramic image. The two models for estimating angle $\varphi$ (Eqs. (1.6) and (1.8)) differ from one another: the first one is linear, while the second one is not. But since we use cameras with the maximal horizontal 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 $\varphi$ that the difference is very small, i.e. the biggest difference is lower than $0.1°$. The experiment in Sec. 1.8.6 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 $\theta_0$ and the radius $r$ in relation with the one-pixel error in estimation of the angle $\varphi$ in the end of Sec. 1.7.4.

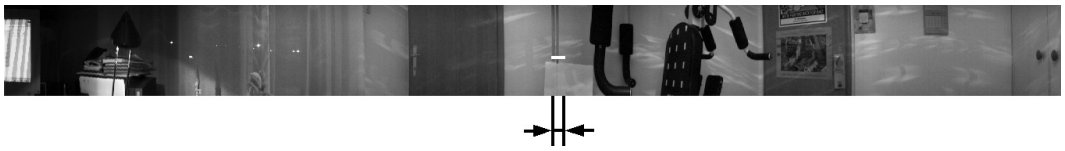
### 1.7.3 Constraining the search space on the epipolar line



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 1.7: We can effectively constrain 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. 1.7a). 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.

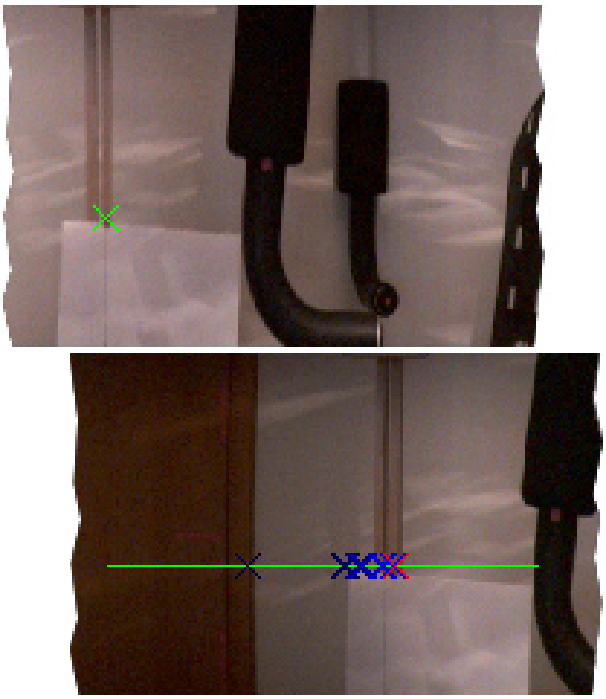


Figure 1.8: Constraining the search space on the epipolar line in case of $2\varphi = 29.9625°$. In the left eye panorama (top image) we have denoted the point for which we are searching the corresponding point with a green cross. In the right eye panorama (bottom image) we have used green color to mark 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.

From Eq. (1.5) 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 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. 1.4) 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 $\theta_0$ 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 $\theta_0$ 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 $\varphi$:

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

2. Theoretically, the estimation of depth is not constrained upwards, but from Eq. (1.5) 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.

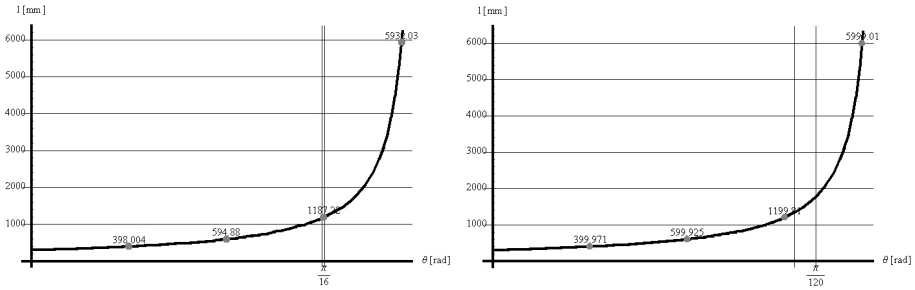
   Equivalently, like in case of the minimal possible depth estimation $l_{\min}$, the maximal possible depth estimation $l_{\max}$ also depends upon the value of the angle $\varphi$:

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

   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 $\varphi \text{ div } \frac{\theta_0}{2} = 145$ pixels in case of $2\varphi = 29.9625^\circ$ (Figs. 1.7b and 1.8) and only 17 in case of $2\varphi = 3.6125^\circ$ (Fig. 1.7c).

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




a) $2\varphi = 29.9625°$ b) $2\varphi = 3.6125°$

Figure 1.9: The dependence of depth $l$ on angle $\theta$ (Eq. (1.5), $r = 30$ cm and two different values of $\varphi$ are used). To visualize the one-pixel error in estimation of the angle $\theta$, we have marked the interval of width $\frac{\theta_0}{2} = 0.102857°$ between the vertical lines near the third point.

### 1.7.4 Meaning of the one-pixel error in estimation of the angle $\theta$

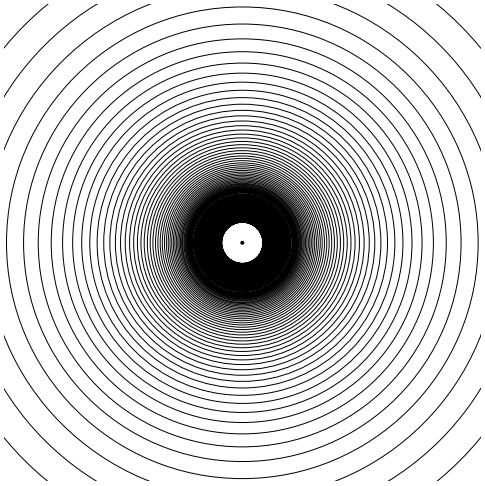
Let us first define what we mean under the term one-pixel error. As the images are discrete, we would like to know what is the value of the error in the depth estimation if we miss the right corresponding point for only one pixel. And we would like to have this information for various values of the angle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

Before we illustrate the meaning of the one-pixel error in estimation of the angle $\theta$, let us take a look at the graphs in Fig. 1.9. The graphs show the dependence of the depth function $l$ on the angle $\theta$ when two different values of the angle $\varphi$ are used. It is evident that the depth function $l$ rises slower in case of a bigger angle $\varphi$. This property decreases the error in the depth estimation $l$ when a bigger angle $\varphi$ 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. 1.9). If we compare the width of the interval in both graphs with respect to the width of the interval that $\theta$ 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 $\varphi$ is used. This subsequently means that the one-pixel error in estimation of the angle $\theta$ is much smaller when a bigger angle $\varphi$ is used, since a shift for the angle $\theta_0$ describes the shift of the camera for a single column of pixels.

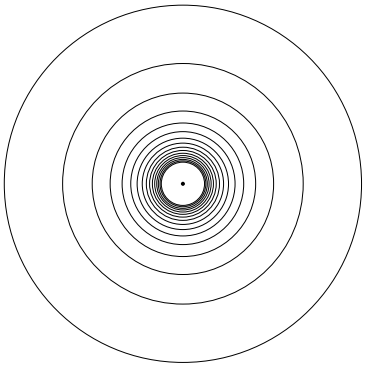
Because of a discrete horizontal axis $\theta$ (Fig. 1.9), with intervals $\frac{\theta_0}{2}$ degrees wide (in our case $\theta_0 = 0.205714°$), the number of possible depth estimates is proportional to the angle $\varphi$: we can calculate $\varphi$ div $\frac{\theta_0}{2} = 145$ different depth values (Eq. (1.5)) if we use the angle $2\varphi = 29.9625°$ (Fig. 1.10a) and only 17 different depth values if we use the angle $2\varphi = 3.6125°$ (Fig. 1.10b). This is the disadvantage of small angles $\varphi$ (see the experiment in Sec. 1.8.1).

Let us illustrate the meaning of the one-pixel error in estimation of the angle $\theta$: We would like to know what is the error of the angle $\theta$ at the beginning of the interval over which $\theta$ is defined ($\theta \in [0, \varphi]$) and what is the error of the angle $\theta$ 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 $\varphi$. In this example we use our already



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



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

Figure 1.10: The number of possible depth estimates is proportional to the angle $\varphi$. Each circle denotes a possible depth estimation value.

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

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

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

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

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

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

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

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

Table 1.2: The one-pixel error $\Delta l$ in estimation of the angle $\theta$, where $r = 30$ cm (Eq. (1.5)).

standard values for the angle $\varphi$: $2\varphi = 29.9625°$ and $2\varphi = 3.6125°$. The results in Tab. 1.2 give the values of the one-pixel error in estimation of the angle $\theta$ for different values of parameters $\theta$ and $\varphi$.

From the results in Tab. 1.2 we can conclude that the error is much bigger in case of a smaller angle $\varphi$ than in case of a bigger angle $\varphi$. The second conclusion is that the value of the error increases as the value of the angle $\theta$ gets closer to the value of the angle $\varphi$. This is true regardless of the value of the angle $\varphi$. This two conclusions are also evident from Fig. 1.10: 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. 1.8.3). The figure nicely illustrates the fact that in case of a small angle $\varphi$, we can estimate only a few different depths and the fact that the one-pixel error in estimation of the angle $\theta$ 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 $\varphi$. 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 $\theta_0$ and $r$ we change the size of the error:

- By increasing the resolution of captured images, we decrease the angle $\theta_0$ (Eq. (1.10)) 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 $\Delta l$, while the reconstruction process takes more (less) time by nearly the same factor. By decreasing (increasing) the value $\theta_0$ we are able to calculate more (less) depth values and consequently, we achieve bigger (lower) accuracy. Another way to influence the parameter $\theta_0$ is to vary the horizontal view angle $\alpha$. This influence is presented separately in Sec. 1.7.7.

- 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 $\Delta l$. 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 $\Delta l$ is too big. If we vary the parameter $r$, the number of possible depth estimates naturally stays the same.

### 1.7.5 Definition of the maximal reliable depth value

In Sec. 1.7.3 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 $\theta$ for these two estimated depths. Let us examine the size of the error $\Delta l$ 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 $\theta$ 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 $\theta$ 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 1.3: The one-pixel error $\Delta l$ in estimation of the angle $\theta$ for the minimal possible depth estimation $l_{\min}$ and the maximal possible depth estimation $l_{\max}$ with respect to the angle $\varphi$ and the radius $r$=30 cm.

In Tab. 1.3 we have gathered the error sizes for different values of the angle $\varphi$. The results confirm statements in Sec. 1.7.4. 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 $\varphi$. 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 ($\Delta l$) for a single pixel in the estimation of the angle $\theta$. Using it, we subsequently define the maximal reliable depth value (see the example in the next section).

### 1.7.6 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. (1.5)) and with (Eq. (1.9)) 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 $\Delta l$ 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 $\beta$ into Eq. (1.9):

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

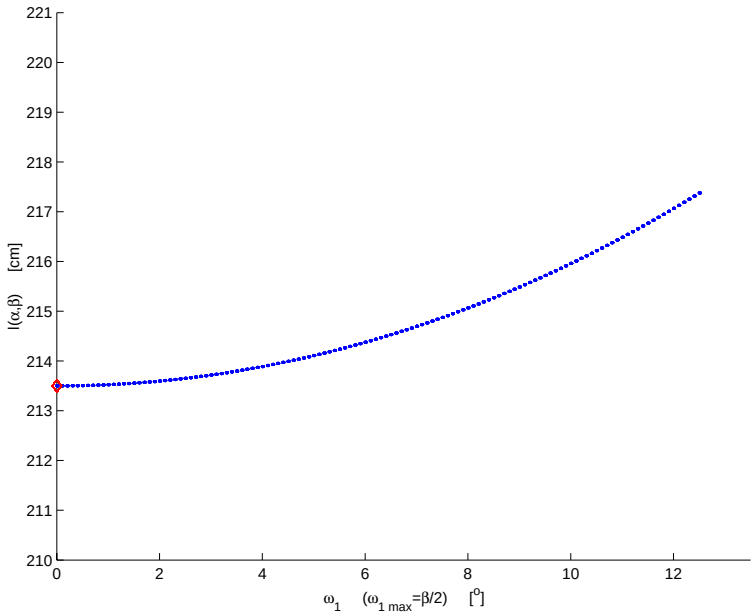



Figure 1.11: The contribution of the vertical reconstruction is small for the camera with the horizontal view angle $\alpha = 34°$ and the vertical view angle $\beta = 25°$ (Eq. (1.9)). The diamond marks the depth $l_{\max}$ estimated without addressing the vertical reconstruction (Eq. (1.5)). For detailed description see Sec. 1.7.6.

we get $l_{\max}(\alpha, \beta) = 217.4$ cm (Fig. 1.11). 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. 1.8.4. By increasing (decreasing) the angle $\beta$ (using different cameras) we also increase (decrease) the contribution of the vertical reconstruction.

### 1.7.7 Influence of using different cameras

Each camera can be characterized by its horizontal view angle $\alpha$. According to Eq. (1.10), the angle $\theta_0$ gets bigger with bigger $\alpha$, 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 $\alpha$ 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 $\alpha$ (we used different cameras). In Fig. 1.12 we can see that for the cameras mentioned in the end of Sec. 1.4, 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

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

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

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

Figure 1.12: Different cameras characterized by the horizontal view angle $\alpha$ give panoramic images with different horizontal resolution $W_{pan}$.

width $W_{pan} = 1501$ pixels and

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

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. 1.7.4, by varying the resolution of the captured images we vary the horizontal and the vertical resolution of the generated panoramic image at fixed $\alpha$. But now we can add one more conclusion, namely, if we vary $\alpha$ 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 $\varphi_{\max} = \alpha/2$, $\theta_0$ 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 ($\alpha$) 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 $\theta_0$ ($\theta_0(\alpha)$, $\theta_0(\varepsilon)$) in Sec. 1.7.2).) This means that the comparison of results gained using different cameras should not be done at the similar $\varphi$, but rather at the similar number of the possible depth estimates. This fact is used in the experiment in Sec. 1.8.8.

The size of the one-pixel error $\Delta l$ in estimation of the angle $\theta$ (Sec. 1.7.4), for $\varphi$'s defined in described way, is also similar. This is evident from Tab. 1.4.

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

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

| | | $\theta$ | $\theta + \frac{\theta_0}{2}$ |
|---|---|---|---|
| $l$ [mm] | $2\varphi = 29.9625°$ ($\alpha = 34°$) | 2373.2 | 2507 |
| | $2\varphi = 38.47875°$ ($\alpha = 39.72°$) | 2355.8 | 2488.8 |
| $\Delta l$ [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 1.4: The one-pixel error $\Delta l$ in estimation of the angle $\theta$ for different cameras ($\alpha$) at the similar number of the possible depth estimates following from $\varphi$, where $r = 30$ cm (Eq. (1.5)). For $2\varphi = 29.9625°$ the results are the same as in Tab. 1.2. Consequently, the same $\theta$ values are used in this table.

## 1.8 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 $160 \times 120$ 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. (1.5)) or $l(\alpha, \beta)$ (Eq. (1.9))) 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. 1.8.1, 1.8.2 and 1.8.3) we use $l$ (Eq. (1.5)) rather than $l(\alpha, \beta)$ (Eq. (1.9)), so that afterwards we are able to demonstrate the influence of the vertical reconstruction on the reconstruction accuracy.

### 1.8.1 Influence of different $\varphi$ 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 $\varphi$ in Sec. 1.7.4. The results were obtained with *camera #1.*
**Results:** The comparison of results using $2\varphi = 3.6125°$ and $2\varphi = 29.9625°$ (see Sec. 1.7.1 about how these values were obtained) is presented in Tab. 1.5.

| 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% | |

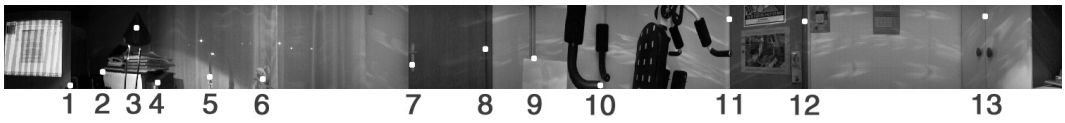



Table 1.5: The comparison of results for two different values of $\varphi$.

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

### 1.8.2 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. 1.6.

| | 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 1.6: 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 $\varphi$, 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 $\varphi$ we have to check only 17 pixels on the epipolar line, while in case of bigger angle $\varphi$ 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 $2\varphi = 29.9625^\circ$ 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 $2\varphi = 29.9625^\circ$ 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. 1.7.7, 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.

### 1.8.3 Influence of different $\varphi$ 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. 1.13. 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*.

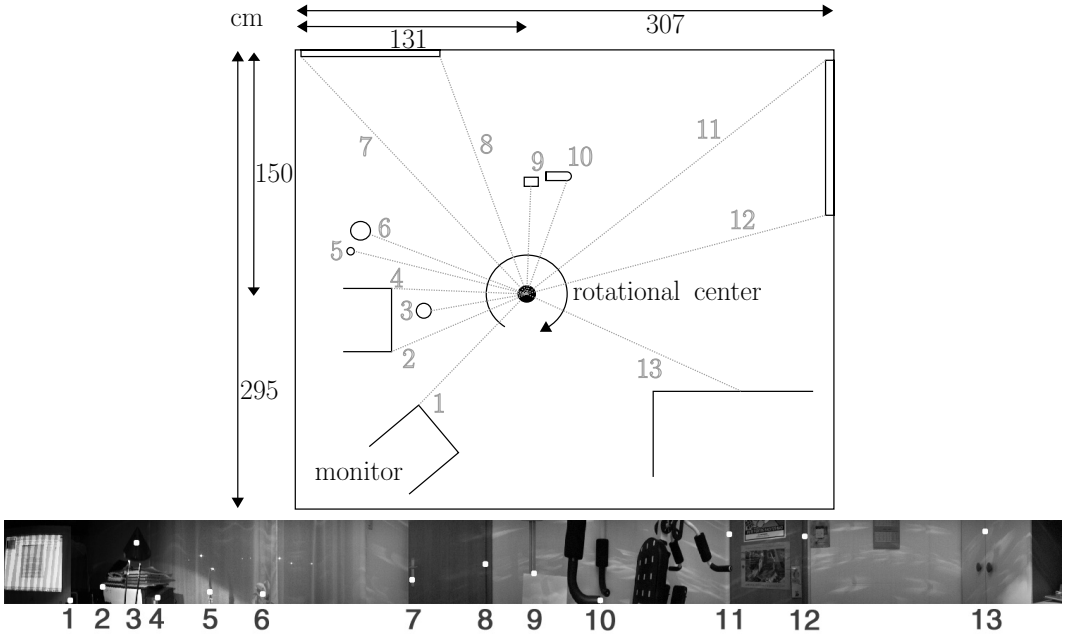
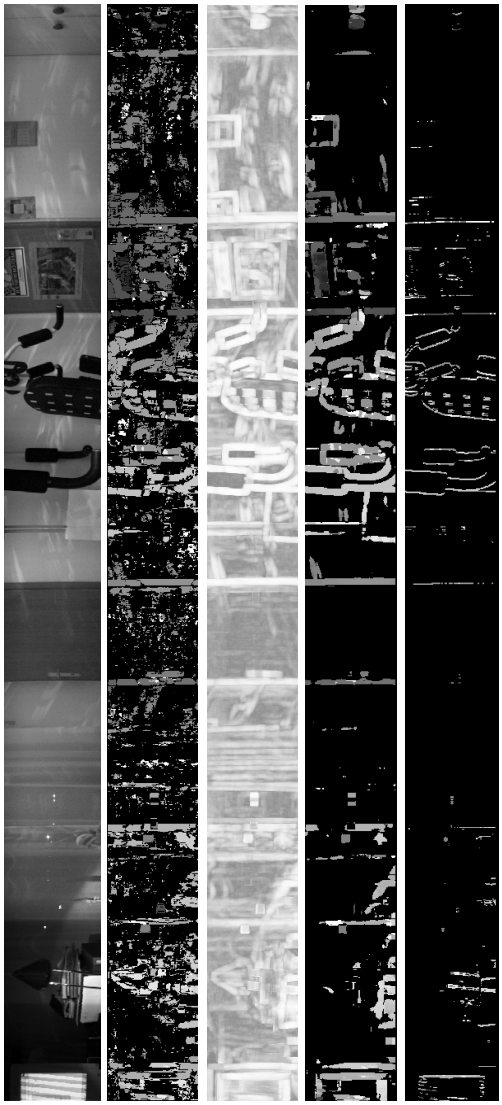



Figure 1.13: The top picture contains the plan of the reconstructed room. In the bottom picture we have marked the features on the scene that help us evaluate the quality of generated depth images.

**Results:** Fig. 1.14 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.

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



a) b) c) d) e)

Figure 1.14: 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.

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. 1.15, 1.16, 1.17 and 1.18:

- 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. 1.15 for the angle $2\varphi = 29.9625°$ and in Fig. 1.16 for the angle $2\varphi = 3.6125°$. We have used back-correlation and weighting. In Figs. 1.15 and 1.16 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. 1.17 and 1.18 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. 1.17 and 1.18 the results have been derived from the sparse depth image gained by using back-correlation. The result in Fig. 1.17 is given for the angle $2\varphi = 29.9625°$ and the result in Fig. 1.18 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. 1.7 — and this was exactly the point of this experiment.

In Fig. 1.16 we can observe two properties of the system (Sec. 1.7.4): 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 $\varphi$, we can estimate only a few different depths and the fact that the one-pixel error in estimation of the angle $\theta$ 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.

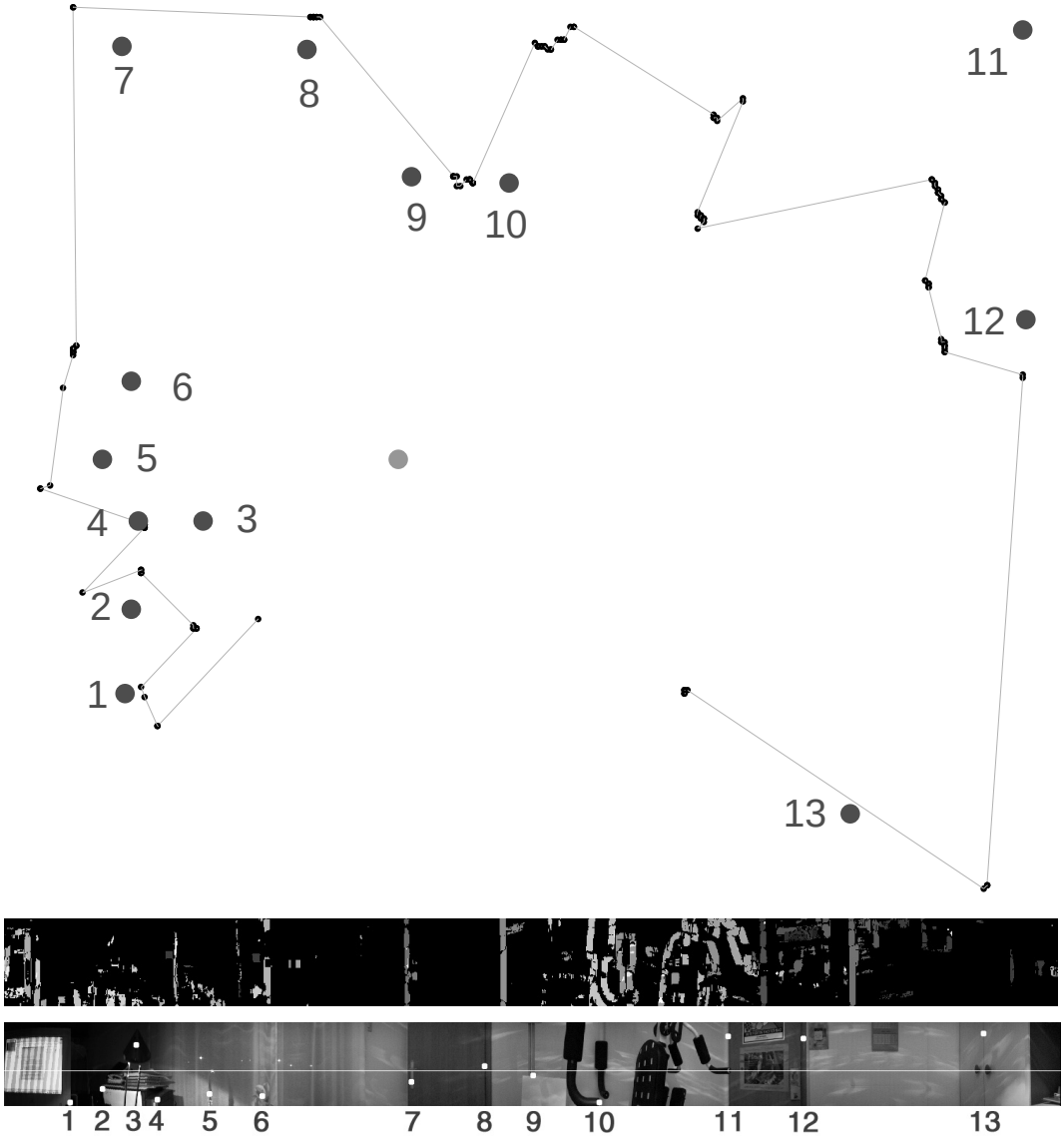



Figure 1.15: The top picture is a ground-plan showing the result of the reconstruction process based on the 68th row of the dense depth image. We have used back-correlation and weighting for the angle $2\varphi = 29.9625°$. The corresponding depth image is shown in the middle picture. For better orientation, 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.

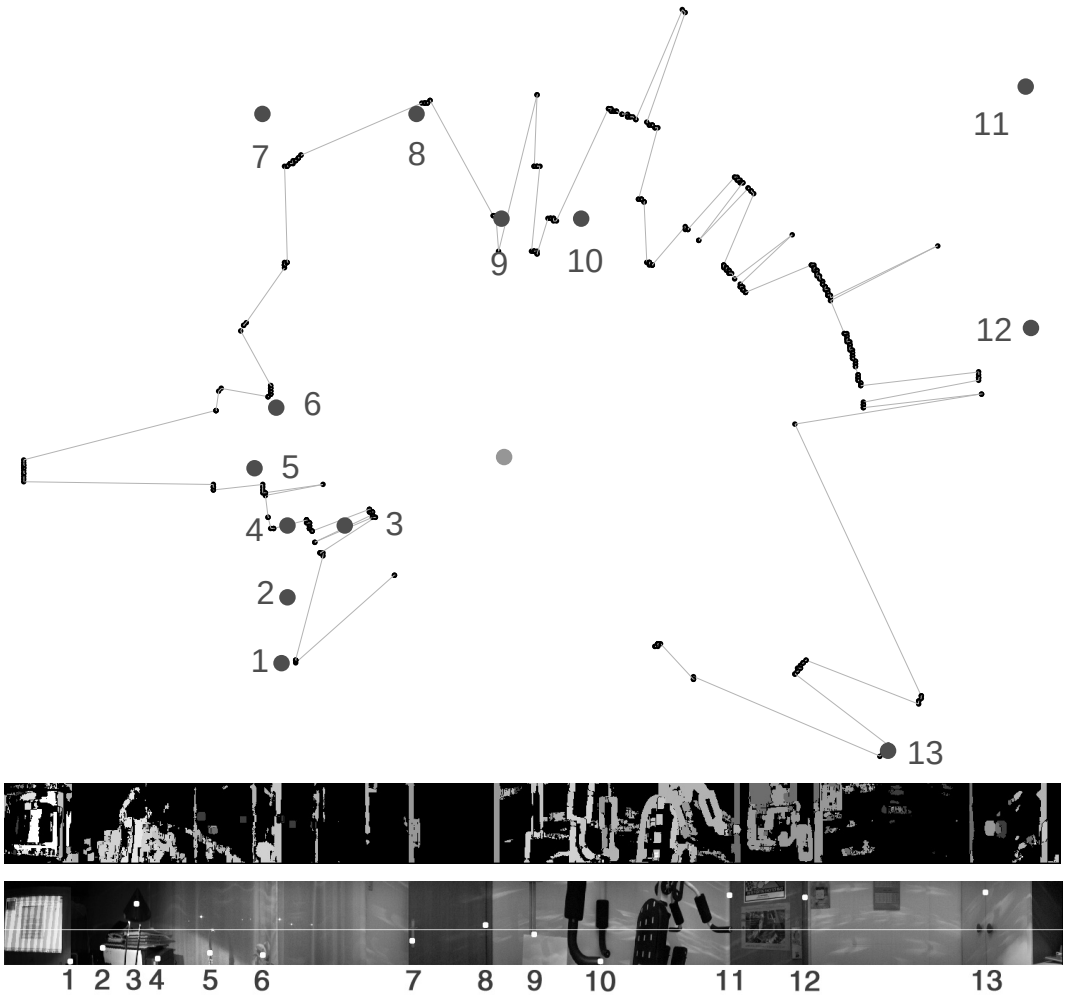



Figure 1.16: The top picture is a ground-plan showing the result of the reconstruction process based on the 68th row of the dense depth image. We have used back-correlation and weighting for the angle $2\varphi = 3.6125°$. The corresponding depth image is shown in the middle picture. For better orientation, 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.

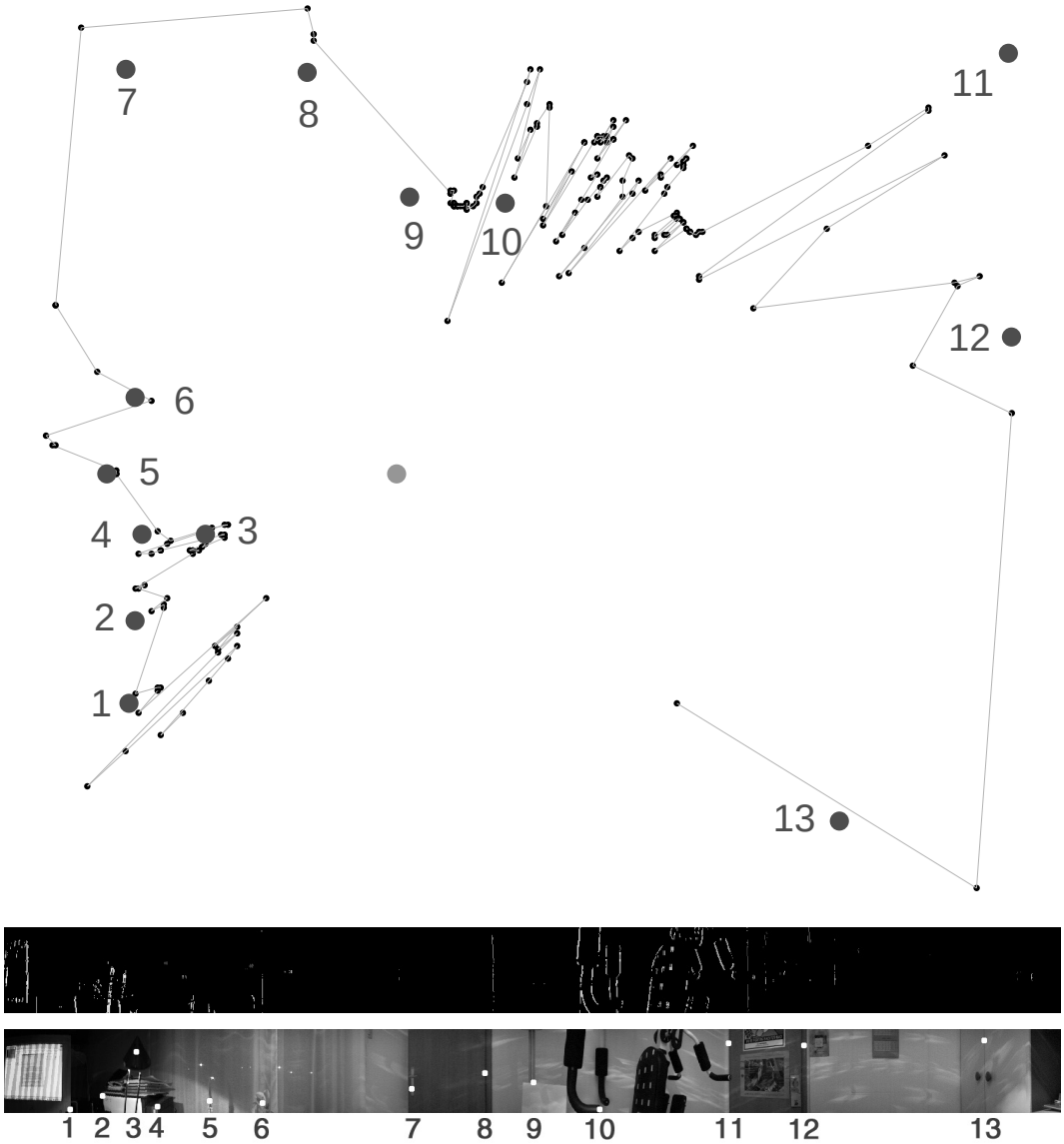



Figure 1.17: The top picture is a ground-plan showing the result of the reconstruction process based on the average depth within each column of the sparse depth image. We have used back-correlation for the angle $2\varphi = 29.9625°$. The corresponding sparse depth image is shown in the middle picture.

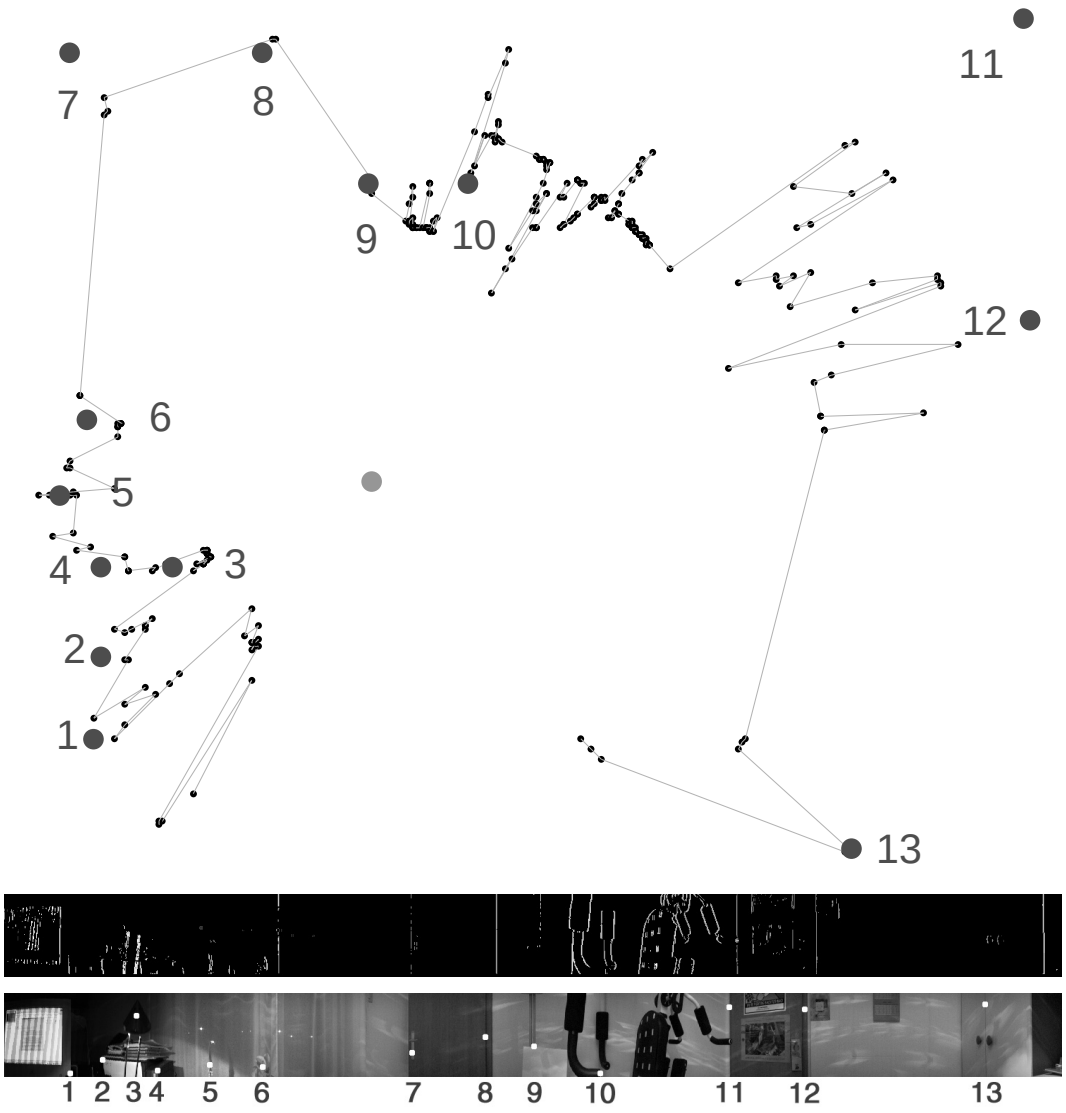



Figure 1.18: The top picture is a ground-plan showing the result of the reconstruction process based on the average depth within each column of the sparse depth image. We have used back-correlation for the angle $2\varphi = 3.6125°$. The corresponding sparse depth image is shown in the middle picture.

### 1.8.4 Influence of addressing the vertical reconstruction

**Experiment background:** In Sec. 1.7.6 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. (1.5)) and $l(\alpha, \beta)$ (Eq. (1.9)) is presented in Tab. 1.7.

| 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% | |

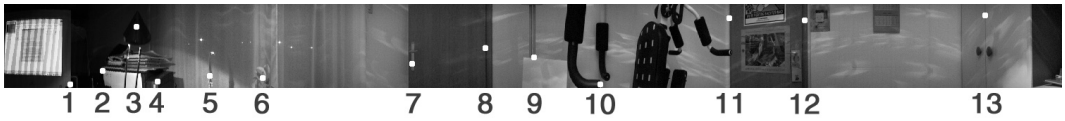



Table 1.7: The comparison of results without and with addressing the vertical reconstruction.

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

### 1.8.5 Influence of different $\theta_0$ values on the reconstruction accuracy

**Experiment background:** See the discussion on estimation of the angle $\theta_0$ ($\theta_0(\alpha)$, $\theta_0(\varepsilon)$) in Sec. 1.7.2. The results were obtained with *camera #1*.
**Results:** The comparison of results using $\theta_0(\alpha) = 0.2125°$ (Eq. (1.10)) and $\theta_0(\varepsilon) = 0.205714°$ (the estimation based on the accuracy of our rotational arm) is presented in Tab. 1.8.

| feature | $d$ [cm] | $\theta_0(\varepsilon) = 0.205714°$ | | $\theta_0(\alpha) = 0.2125°$ | |
|---|---|---|---|---|---|
| | | $l(\alpha,\beta)$ [cm] | $l(\alpha,\beta) - d$ [cm (% of $d$)] | $l(\alpha,\beta)$ [cm] | $l(\alpha,\beta) - d$ [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% | |

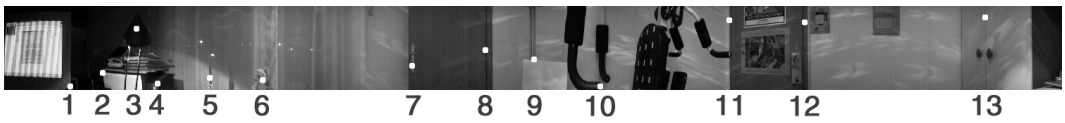



Table 1.8: The comparison of results for two different values of $\theta_0$.

**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.

### 1.8.6 Linear versus non-linear model for estimation of angle $\varphi$

**Experiment background:** See the discussion on estimation of the angle $\varphi$ in Sec. 1.7.2. The results were obtained with *camera #1*.
**Results:** The comparison of results using $2\varphi = 29.9625°$ (Eq. (1.6)) and $2\varphi = 30.15774565°$ (Eq. (1.8)) is presented in Tab. 1.9.

| feature | $d$ [cm] | $2\varphi = 29.9625°$ $l(\alpha,\beta)$ [cm] | $2\varphi = 29.9625°$ $l(\alpha,\beta) - d$ [cm (% of $d$)] | $2\varphi = 30.15774565°$ $l(\alpha,\beta)$ [cm] | $2\varphi = 30.15774565°$ $l(\alpha,\beta) - d$ [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% | |

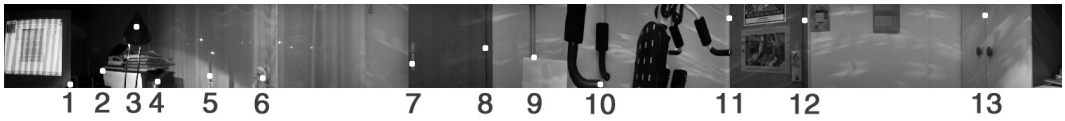



Table 1.9: The comparison of results for two different values of $\varphi$: the first one is gained from the linear and the second one from the non-linear model for estimation of angle $\varphi$.

**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. 1.8.8): 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.

### 1.8.7 Repeatability of results — Different room

**Experiment background:** We want to see if we can achieve similar results as in Sec. 1.8.4, using the same camera (*camera #1*) in a different room?
**Results:** The results obtained in the different room are presented in Tab. 1.10.

| 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%) |
| | | $\text{AVG}_{\%}$=4.5% ± 4.5% | |


Table 1.10: The results obtained in the different room, but with the same camera as in Sec. 1.8.4.

**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. 1.7.4).

### 1.8.8 Repeatability of results — Different cameras

**Experiment background:** We want to see if we can achieve similar results as in Secs. 1.8.4 and 1.8.7, using different cameras in a different, third room? As mentioned in Sec. 1.7.7, the comparison of results gained using different cameras should not be done at the similar $\varphi$, 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. 1.11. 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 $\beta$, 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. 1.12). 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% | |

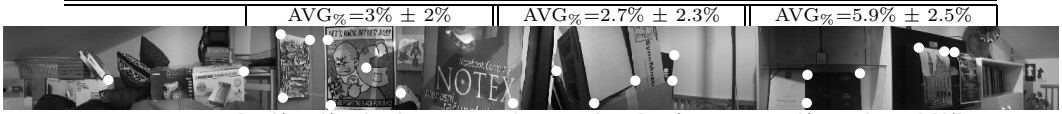



Table 1.11: The results obtained in the third room with three different cameras.

**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.

### 1.8.9 Possibility of systematic error presence in the estimation of $r$

**Experiment background:** In Sec. 1.7.2 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. 1.12 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. 1.7.4),
- the error in estimations of other parameters (e.g. $\alpha$),
- the error due to the lens distortion presence (this matter is addressed in Sec. 1.8.10),
- 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% | |

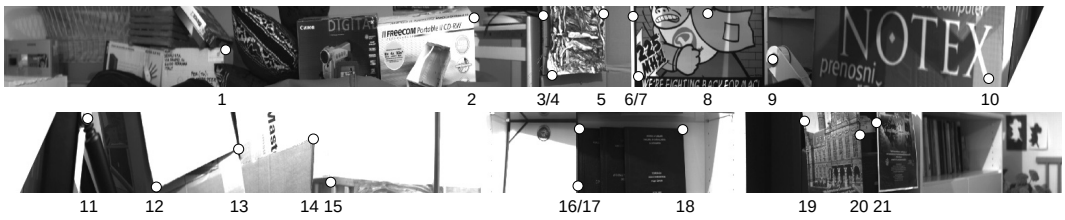



Table 1.12: The comparison of results before and after the optimization of parameter $r$.

### 1.8.10 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. 1.19 nicely illustrates this fact. (The error in 640×480 pixel images is 4 times bigger.)

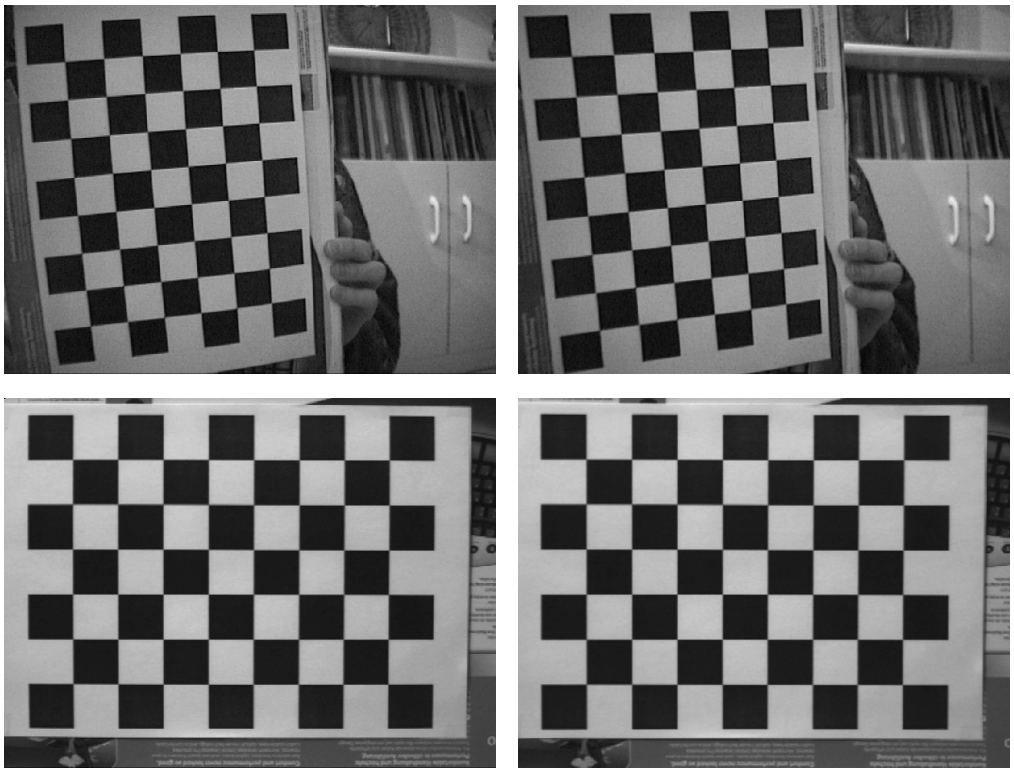


Figure 1.19: Each row of images shows distorted image (left) and undistorted image (right), after the distortion has been suppressed. The images in the top row have been taken with *camera #2*, while the images in the bottom row have been taken with *camera #3*. The resolution of all presented images is 640×480 pixels.

Since the best results obtained with *camera #3* are already very good (Sec. 1.8.9) 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. 1.20 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.

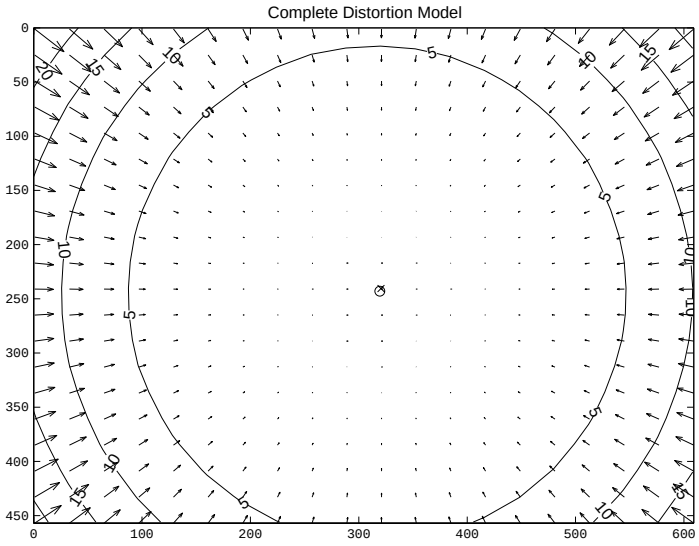



Figure 1.20: The camera model gained after the calibration process over a set of 640×480 pixel images [54]. Note that the values of estimated parameters (that are given in pixels) are 4× smaller for 160×120 pixel images, which we use for generating panoramic images, and that the errors given in the right 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. 1.19). The new vertical field of view $\alpha_{new}$ 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. (1.6)):

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

$W$ is again the width of the captured image.

We also know that different $\alpha$ brings different $r$ (Sec. 1.7.2), 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. (1.2). But $\theta_0$ 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. 1.13. For comparison, on the left side the results using distorted sequence are presented. In case of undistorted sequence, we have used $2\varphi = 37.04933695^\circ$ as it ensures a similar number of possible depth estimates as the basic settings of *camera #2* (Sec. 1.7.7). 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. 1.19). 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% | |

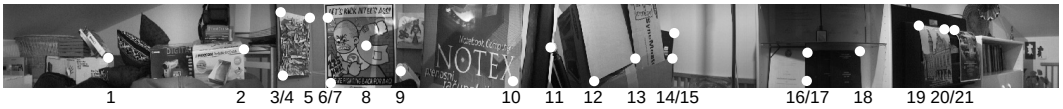



Table 1.13: The comparison of results obtained without and with the lens distortion correction.

## 1.9 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. 1.7 and 1.8. Nevertheless, let us summarize the main conclusions made throughout the chapter and indicate in which section of the chapter each conclusion has been made:

- The geometry of capturing multiperspective panoramic images can be described with a pair of parameters $(r, \varphi)$ (Sec. 1.4). 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. 1.7.1).
- 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. 1.5).
- 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. 1.7.2, 1.8.5, 1.8.6 and 1.8.9).
- We can effectively constrain the search space on the epipolar line (Sec. 1.7.3). This follows directly from the interpretation of the equation for depth estimation $l$ (Eq. (1.5)), 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 $\Delta l$ gets bigger) and 2) the bigger the value $\varphi$ for each camera $(\alpha)$, the bigger the number of possible depth estimates and consequently the bigger the confidence (Secs. 1.7.4, 1.8.1 and 1.8.3).
- We can influence the parameter $\theta_0$ by varying the resolution of captured images or by varying the horizontal view angle $\alpha$ (Secs. 1.7.4 and 1.7.7).
- By varying the radius $r$, we vary the biggest possible and sensible depth estimation $l$ and the size of the one-pixel error $\Delta l$ (Sec. 1.7.4).
- The bigger the value $\alpha$, the smaller the horizontal resolution of panoramic images at fixed resolution of captured images (Sec. 1.7.7). 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 $\Delta l$ (Sec. 1.7.5).

- The contribution of the vertical reconstruction is small in general, but has a positive influence on the overall results (Secs. 1.7.6 and 1.8.4).
- The numbers of possible depth estimates are very similar for different cameras ($\alpha$) at fixed resolution of the captured images (Sec. 1.7.7).
- The size of the one-pixel error $\Delta l$ is also similar at similar number of possible depth estimates for different cameras (Sec. 1.7.7).
- The reconstruction process can execute in real time (Sec. 1.8.2).
- The reconstructed points lie on concentric circles centered in the center of rotation and the distance between circles (the one-pixel error $\Delta l$) increases the further away they lie from the center (Secs. 1.7.4 and 1.8.3).
- The linear model for estimation of angle $\varphi$ have been proved better for a given set of parameters in comparison to the non-linear model (Sec. 1.8.6).
- 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. 1.8.7 and 1.8.8).
- The remaining error in accuracy could be attributed to a number of possible reasons (Sec. 1.8.9).
- 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. 1.8.10).

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.

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. 1.7.2) or the method for defining the maximal reliable depth value (Sec. 1.7.5).

In the end, let us write only the conclusion of all conclusions, which answers to the question written at the very beginning of this chapter (Sec. 1.1.1): Can the system be used for robot localization and navigation in a room? According to the accuracy achieved the answer is: Yes!

Our future work is directed primarily in the development of an application for the real time autonomous localization and navigation of a mobile robot in a room.

**Acknowledgements**

A preliminary and much shorter version of this chapter was published in Kluwer's International Journal of Computer Vision [47].

This work was supported by the Ministry of education, science and sport of Republic of Slovenia (project Computer vision P2-0214).

# 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.