Real time panoramic depth imaging from multiperspective panoramas using standard cameras

Peter Peer*, Franc Solina

University of Ljubljana
Faculty of Computer and Information Science
Tržaška 25, SI-1000 Ljubljana, Slovenia
Tel. +386 1 4768 878, Fax +386 1 4264 647
E-mail: peter.peer@fri.uni-lj.si

Abstract

Recently we have presented a system for panoramic depth imaging with a single standard camera. The 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 from a single sweep around the rotational center, which enables the stereo reconstruction. One of the problems of such a system is the fact that we cannot generate a stereo pair of images in real time. This chapter presents a possible solution to this problem, which is based on simultaneously using many standard cameras. We perform simulations on real scene images to establish the quality of new sensor results in comparison to results obtained with the old sensor. The goal of the chapter is to reveal whether the new sensor can be used for real time capturing of panoramic depth images and consequently for autonomous navigation of a mobile robot in a room. In particular, we focus on the real time generation of panoramic stereo pairs since the calculation of depth images can already be run in real time. The basic panoramic depth imaging system and its real time extension are comprehensively analysed and compared. The analyses reveal a number of interesting properties of the systems. According to the basic system accuracy we definitely can use the system for autonomous robot localization and navigation tasks. The assumptions made in the real time extension of the basic

*Corresponding author

system are proved to be correct, but the accuracy of the new sensor generally deteriorates in comparison to the basic sensor.

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

1 Introduction

Real time panoramic depth imaging from multiperspective panoramas is an issue that is not well covered in the literature. There have been discussions [11] about it, but nothing has been done in practice so far, at least not by using the idea of multiperspective panoramas (the definition is given in the beginning of the next section).

Generally, mosaic-based procedures for building panoramic images [1, 3, 5, 11, 12, 17, 18, 21] can be marked as non-central, they do not execute in real time and they give high resolution results. Thus mosaicing is not appropriate for capturing dynamic scenes. The main advantage of these procedures over other panoramic imaging systems (like catadioptric systems [25]) is the ability to generate high resolution results. But high resolution results are essential for effective depth recovery based on the stereo effect.

In this chapter we focus on real time generation of a multiperspective panoramic stereo pair. Calculation of depth images from such stereo pairs already runs in real time [24].

In [12] we presented a system for panoramic depth imaging with a single standard camera. We reported a number of interesting properties of the system. In Sec. 3 we summarize them and report a number of new properties and experimental results [13, 14], all in order that the reader gets a good picture about the system. Our main motivation was to establish the quality of results for autonomous robot localization and navigation tasks. In this chapter we go a step further and evaluate a real time extension of the system. To do that, we first in Sec. 4 discuss how the time needed for the generation of a symmetric panoramic stereo pair can be dramatically reduced. In Sec. 5 we go even further and explain how we can achieve real time execution. Sec. 6 presents the depth reconstruction equation for the new setup. The epipolar constraint is discussed in Sec. 7. The evaluation of results is given in Sec. 8. We end the chapter with conclusions in Sec. 9. But let us first start by saying a few words about related work.

2 Related work

One way to build panoramic images is by taking one pixel column out of a captured image and mosaicing the columns. Such panoramic images are called multiperspective panoramic images [21]. 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.

However, multiperspective panoramic images are not something new to the vision community [21]: they are a special case of *multiperspective panoramic images for cel animation* [28], a special case of *crossed-slits (X-slits) projection* [2, 7, 29], they are very similar to images generated by a procedure called *multiple-center-of-projection* [19], by the *manifold projection* procedure [17] and by the *circular projection* procedure [15, 16]. The principle of constructing multiperspective panoramic images is also very similar to the *linear pushbroom camera* principle for creating panoramic images [8].

The papers [11, 21, 22], which are closest to our work about the panoramic depth imaging with a single standard camera [12, 13, 14] 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 the next section is on these two issues. While in [11] 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 [21] used an upgraded *plane sweep stereo* procedure. A key idea behind the approach in [22] is that it enables optimizing the input to traditional computer vision algorithms for searching the correspondences in order to produce superior results.

There are of course other somehow similar possibilities to acquire panoramic depth images. Let us briefly present the system called SOS (Stereo Omnidirectional System) [20, 26, 27]. The SOS system uses standard cameras and produces panoramic depth images in real time, but here all the similarities with our system end. The system consists of 20 stereo units and each unit consists of 3 standard cameras. The units are arranged on each plane of a regular icosahedron. The basic principle for generating panoramic depth images is as follows: Each unit captures three images from which a normal (not panoramic) depth image is computed on a computer with more CPU units. Then the generated depth images are registered into the panoramic depth image. Obviously, the authors use a different concept than we do: they first build standard depth images and then the panoramic depth image, while we first build panoramic images and then the panoramic depth image. So, from this point on, we are interested only in panoramic depth imaging from multiperspective panoramas.

In order to capture stereo panoramic images of dynamic scenes Peleg et al. [18] presented a theory for construction of a special mirror and lens such that viewing the scene through this mirror or lens creates the same rays as those used with the rotating cameras. To our knowledge the systems are still not constructed, while, according to authors, a lot of practical issues need to be solved before a camera is built.

As mentioned, the real time extension of our system to our knowledge has not been suggested and evaluated before.

3 About the basic system

Our basic system for generating panoramic depth images with a single standard camera captures a stereo pair of images while the camera rotates around the center of the system in a horizontal plane. The motion parallax effect which enables the reconstruction can be captured because of the offset of the cameras' optical center from the systems' rotational center (see the small photograph within Fig. 1). The camera is moving around the rotational center in angular steps corresponding to one vertical pixel column of the captured standard image. A symmetric pair of panoramic stereo images are generated so that one column on the right side of the captured image contributes to the left eye panoramic image and the symmetric column on the left side of the captured image contributes to the right eye panoramic image. 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 image row epipolar constraint [9, 10, 21, 23] and minimal reconstruction error [22]).

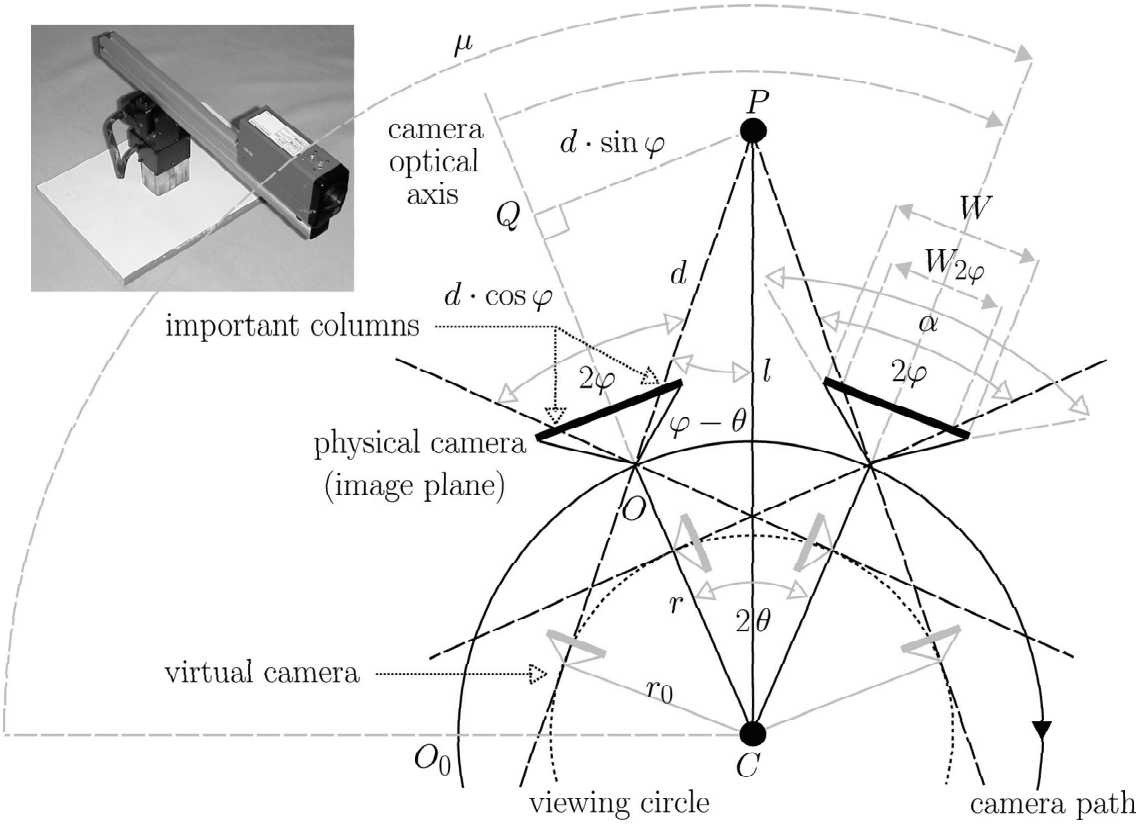

Figure 1: Geometry of our basic system for constructing multiperspective panoramic images. Note that a ground-plan is presented; the viewing circle extends in 3D to the viewing cylinder. The optical axis of the camera is kept horizontal. In the small photograph the hardware part of the system is shown.

The geometry of our basic system for creating multiperspective panoramic images is shown in Fig. 1. The panoramic images are then used as the input to create panoramic depth images. Point C denotes the system's rotational center around which the camera is rotated. The offset of the camera's optical center from the rotational center C is denoted as r, describing the radius of the circular path of the camera. The camera is looking outward from the rotational center. The optical center of the camera is marked with O. The column of pixels that is sewn in the panoramic image contains the projection of point P on the scene. The distance from point P to point C is the depth l, while the distance from point P to point O is denoted by d. Further, θ is the angle between the line defined by points C and O and the line defined by points C and P. In the panoramic image the horizontal axis represents the path of the camera. The axis is spanned by μ and defined by point C, a starting point O_0, where we start capturing the panoramic image, and the current point O. φ denotes the angle between the line defined by point O and the middle column of pixels of the image captured by the physical camera looking outward from the rotational center (the latter column contains the projection of the point Q), and the line defined by point O and the column of pixels that will be mosaiced into the panoramic image (the latter column contains the projection of the point P). Angle φ can be thought of as a reduction of the camera's horizontal view angle α.

The main conclusions made throughout the analysis are (see [12, 13, 14] for discussion about each item):

- The geometry of capturing multiperspective panoramic images can be described with a pair of parameters (r, φ). By increasing (decreasing) each of them, we increase (decrease) the baseline $(2r_0)$ of our stereo system.
- The stereo pair acquisition procedure with only one standard camera cannot be executed in real time.
- The epipolar constraint in case of symmetric stereo pair of panoramic images, which we use in the reconstruction process, is very simple: epipolar lines are image rows.
- 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.
- We can effectively constrain the search space on the epipolar line. This follows directly from the interpretation of the equation for depth estimation l, 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[1] Δl gets bigger) and 2) the bigger the value φ for each camera (α), the bigger the number of possible depth estimates and consequently the bigger the confidence.

- We can influence the parameter θ_0[2] by varying the resolution of captured images or by varying the horizontal view angle α.
- By varying the radius r, we vary the biggest possible and sensible depth estimation l and the size of the one-pixel error Δl.
- The bigger the value α, the smaller the horizontal resolution of panoramic images at fixed resolution of captured images. Consequently, the number of possible depth estimates per one degree gets lower.
- In practice, from the autonomous robot localization and navigation system point of view, we should define the upper boundary of the allowed one-pixel error size Δl.
- The contribution of the vertical reconstruction[3] is small in general, but has a positive influence on the overall results.
- The numbers of possible depth estimates are very similar for different cameras (α) at fixed resolution of the captured images.
- The size of the one-pixel error Δl is also similar at similar number of possible depth estimates for different cameras.
- The reconstruction process (after the stereo pair has been generated) can be executed in real time.
- The reconstructed points lie on concentric circles centered in the center of rotation and the distance between circles (the one-pixel error Δl) increases the further away they lie from the center.
- The linear model for estimation of angle φ have been proved better for a given set of parameters in comparison to the non-linear model[4].
- We can achieve similar reconstruction accuracy with panoramas build from only one-pixel column of the captured images in different rooms, even with different cameras.

[1] As the images are discrete, we like to know what is the value of the error in the depth estimation if we miss the right corresponding point for one pixel.

[2] Our system works by moving the camera for the angle corresponding to one pixel column of the captured image.

[3] In contrast to [12], we have incorporated the vertical view angle β into the equation for depth estimation l in [13].

[4] The non-linear model contains the focal lenght f explicitly.

- The remaining error in accuracy could be attributed to a number of possible reasons (e.g. to the fact that we are limited with the number of possible depth estimates, which are approximations of the real distances etc.).
- 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.
- According to the basic system accuracy, we definitely can use it in autonomous robot localization and navigation tasks.

This system however cannot generate panoramic stereo pair in real time. To illustrate this fact, we can write down the following example from practice: if the system builds a panoramic stereo pair from standard images with resolution of 160×120 pixels, using a camera with the horizontal view angle $\alpha = 34°$, it needs around 15 minutes to complete the task.

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

4 Building panoramic images from wider stripes

This task is by all means much faster, but at the same time we have to make a compromise between the speed of the capturing task and the quality of the stereo pair. First of all, the wider the stripes are, the more obvious are the stitches between the stripes in the panoramic image. Then, with the wider stripe the difference between the amount of lens distortion of the first and the last pixel columns in the stripe is more noticeable. But the real problem arises from the fact that stripes introduce a property, which influences the coverage of the scene, as presented in the next section. As we show in the experimental results (Sec. 8), we are satisfied with the result when we use 14 pixel columns wide stripes and we think that it represents a good compromise. This statement is naturally highly related to the camera that we use.

In that case, the horizontal view angle of the camera is 34°, where 14 columns represent the angle of approximately 3°, the building process takes approximately 14 times less, i.e. around a minute.

4.1 Property of using stripes

If we observe the panoramic image built from stripes closely, we can notice that the image is not perfect. In this case we are not referring to stitches nor lens distortion. These problems are present, but are not too disturbing. Another problem can be noticed on close objects on the scene, which have a nice texture on them (like text). In such a case we can see that some points on the scene are not captured (Fig. 2).

Figure 2: The wider is the stripe, the more scene points are not captured in the panoramic image: the left panoramic image was built from single columns, while the right panoramic image was built from 14 columns wide stripes. Note how very distant scene points are well captured in both examples and how some nearby scene points (text on the box) are not captured in the second example.

Of course, this is partly because of the fact that we are dealing with the images, which are discrete (for example: we cannot take a half of a pixel), but if we take a look at the geometry of the system (Fig. 1), we can see that this is not the only reason. If we consider two successive steps of the system, we can see that the stripes that contribute to the panoramic image do not cover all the scene.

The drawing in Fig. 3, shows the formation of the panoramic image with respect to the light rays (compare with Fig. 1). The drawing is done for the actual case of horizontal view angle $\alpha = 34°$ to illustrate the problem of uncaptured scene points. The large circle represents the camera path. The small circle represents the viewing circle. Point C in the middle of the circles represents the center of rotation. Point O represents the position of the optical center on the camera path. A middle line going from the optical center O outwards represents the light ray that is in the middle of light rays that form a stripe in each captured image that contributes to the panoramic image. A line tangent to the viewing circle is the extension of that light ray towards the virtual camera. Two lines on each side of this light ray going from the optical center O outwards represent bordering light rays that form a stripe in each captured image that contributes to the panoramic image.

Let us first examine the example given on the right side in Fig. 2, where the shift angle corresponds to the stripe of 14 pixel columns of the captured image (W_s=14): $\alpha = 34°$, $r = 30$ cm, $2\varphi_{\min} = 27.082501°$, $2\varphi_{\max} = 32.842499°$ (these two values $\varphi_{\min}$ and $\varphi_{\max}$ define the stripe, which consists of more pixel columns, similarly as φ defined one pixel column; for more information see Sec. 6), $\theta_0 = W_s \cdot 0.205714°$. The drawing is presented in Fig. 3. Since each stripe of the captured image is formed from more light rays, which also circumscribe a similar angle to the systems' shift angle (compare definitions of $\theta_0(\alpha)$ and $\theta_0(\varepsilon)$ given below), the middle light ray and the border light rays are presented for each stripe that contributes to the panoramic image. We can observe two properties on this drawing (note the zoomed in detail in it): 1) We really cannot capture all the scene points, since there is a gap between two successive stripes that contribute to the panoramic image. 2) With the growing distance from the rotational center the bordering light rays of two successive stripes are getting nearer until they intersect. Generally, the shift angle and the angle corresponding to the stripe of the captured image should be the

same, but they are not due to the limited accuracy of our rotational arm. In this general and ideal case the bordering light rays of two successive stripes would be parallel (for example, imagine that both angles are 90°). Since we deal here with two independent discretizations, namely with the discretization of the standard image (pixels; $\theta_0(\alpha)$) on one side and with the discretization of the rotational arm (the minimal step of the rotational arm defined by its maximal accuracy; $\theta_0(\varepsilon)$) on the other side, it is almost impossible to achieve the ideal case ($\theta_0(\alpha) = \theta_0(\varepsilon)$). Because of these discretizations the bordering light rays of two successive stripes are normally getting nearer ($\theta_0(\alpha) > \theta_0(\varepsilon)$) or further appart ($\theta_0(\alpha) < \theta_0(\varepsilon)$).

Already the first property by itself reveals that there is always a part of the scene which cannot be captured. And since the effect loses on significance with a bigger distance from the rotational center, the distant points on the scene look well captured (Fig. 2), also since the bigger distance to the object results in lower resolution of the object. On the other hand, in our case, the bordering light rays of two successive stripes are getting nearer with bigger distance, which means that this effect is getting even smaller with increased distance. This also means that in the far distance the bordering light rays are intersecting, but since the intersection point is far away the effect of repeatability of scene points in the panoramic image is not noticeable, again because with the bigger distance of the system from the object its resolution in the image gets lower.

A similar effect is present when we build panoramic images from only one pixel column of each captured image, while one pixel width is not infinitesimal, although the part of the panoramic image presented on the left side in Fig. 2 looks faultless. In this case the gaps are much smaller.

So, we can write one more conclusion with respect to this property. Namely, the wider is the stripe, the more scene points are not captured in the panoramic image (Figs. 2 and 3). And this holds regardless of the position of the stripe in the captured image. We can take a stripe from the middle of the captured image or from the edge of it, but the presented property would still be seen on the resulting panoramic image. Naturally, by using single columns we achieve best possible result (Fig. 2), though still not perfect, since the described property still holds, but is not so obvious. By using smaller r the gaps are also smaller, which means that with smaller r we cover more scene points.

Another conclusion is that if we want to capture the parallax effect, we have to accept the fact that not all the scene points are captured. When it comes to scene reconstruction, we can address the missing scene information problem in a similar manner as in the case of a normal, non-panoramic stereo system (e.g. problem of occlusions).

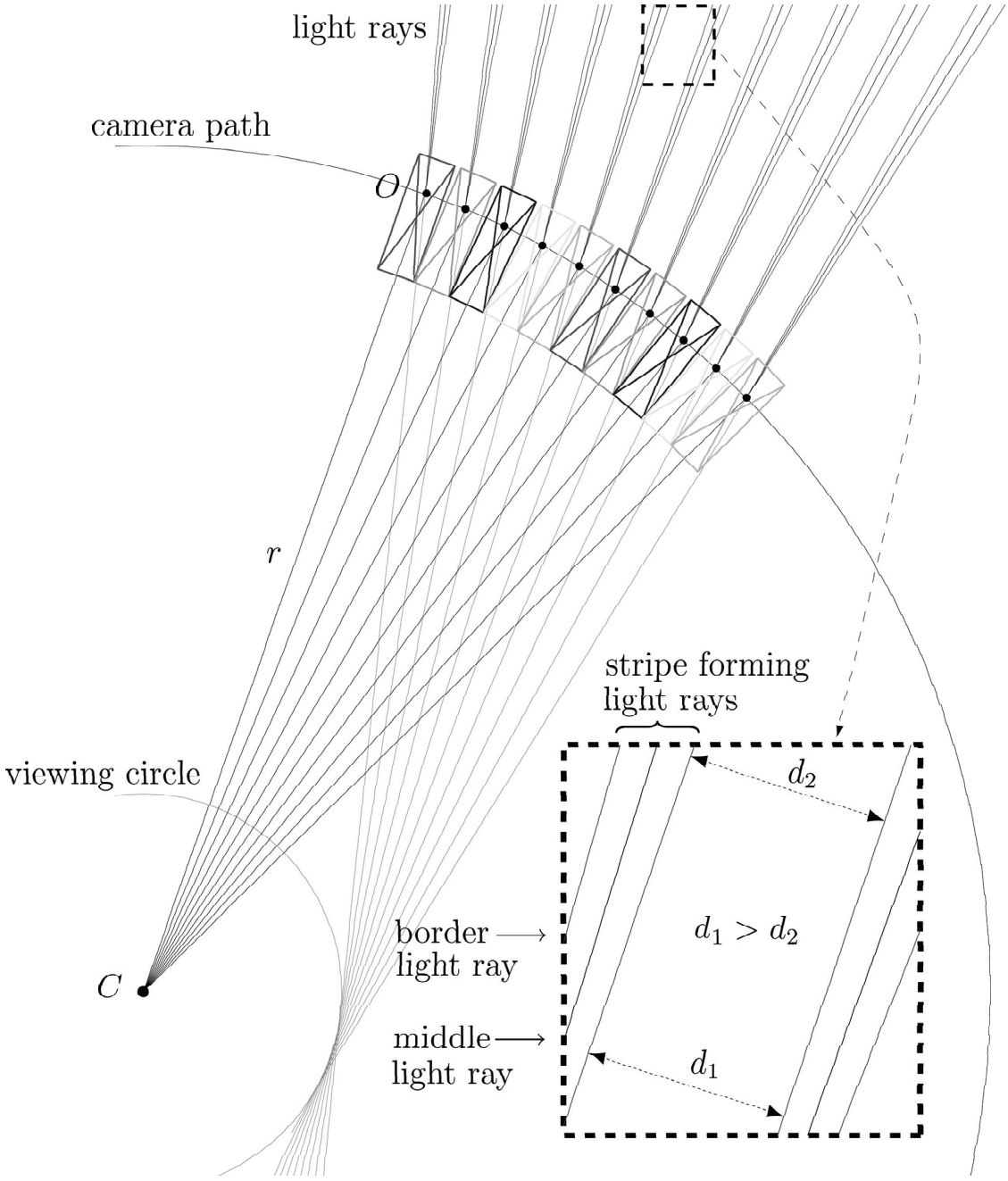

Figure 3: The drawing shows the formation of the panoramic image with respect to the light rays. In this case the shift angle corresponds to 14 pixel columns of the captured image. For detailed description see Secs. 4.1 and 5.

5 Achieving real time

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

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

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

Fig. 3 also shows the drawing of a real time sensor. The boxes with crosses represent the physical cameras positioned with respect to the center of rotation. The sensor is made out of cameras presented in the photograph within Fig. 1 ($\alpha = 34^\circ$), where the angle between successive cameras corresponds to 14 columns of the captured image with the resolution of 160×120 pixels. This camera is probably much to big to be used in a real sensor (the ground plan dimensions of the camera are 14.5×4.5 cm), since r in this case should be set to at least 100 cm. On the other hand, we could set r to 30 cm and this implies that the camera should be smaller for $3.\overline{3}$ times (in this case the ground plan dimensions of the camera should be 4.35×1.35 cm). We have simulated such sensor with our camera in order to determine its accuracy. Namely, nowadays there are many practical solutions how to increase processing power, but before we build a real system, we have to know whether the accuracy of the proposed system is satisfactory.

The experimental results obtained by rotating a single camera are given in Sec. 8, where we have also tested other cameras, different widths of the stripes etc. If we built the sensor, we would normally use board cameras, where the lens, as the biggest part, is even smaller and the board to which the lens is attached is flexible, so that we are able to put the lenses completely together.

6 Stereo reconstruction from stripes

By following the sinus law for triangles, we can simply write the equation for the depth l as (Fig. 1):

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

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

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

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

where dx is the absolute value of the difference between x coordinates of the corresponding points in the left eye panoramic image and in the right eye panoramic image, while θ_0 is the angle corresponding to one pixel column of the captured image and consequently the angle for which we have to move the robotic arm if we build the panoramic images from only one column of the captured image. Using analogy for this equation and having in mind that we are building the panoramic images from stripes, we can write the following equation (Figs. 1 and 4):

$$2\theta = \theta_l + \theta_r. \tag{3}$$

When we use one column instead of stripes then $\theta_l = \theta_r$ (Fig. 1), but this is not necessary true if we use stripes. In general these two values are different, but the property following from the equation

$$\frac{\theta_l}{\theta_r} = \frac{\varphi_l}{\varphi_r} \tag{4}$$

shows that the ratio of these two values is related to the angles φ (Fig. 4). The bigger φ_l gets, the bigger gets the corresponding θ_l. Now we can simply express θ_r and θ_l from Eqs. (2), (3) and (4) as:

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

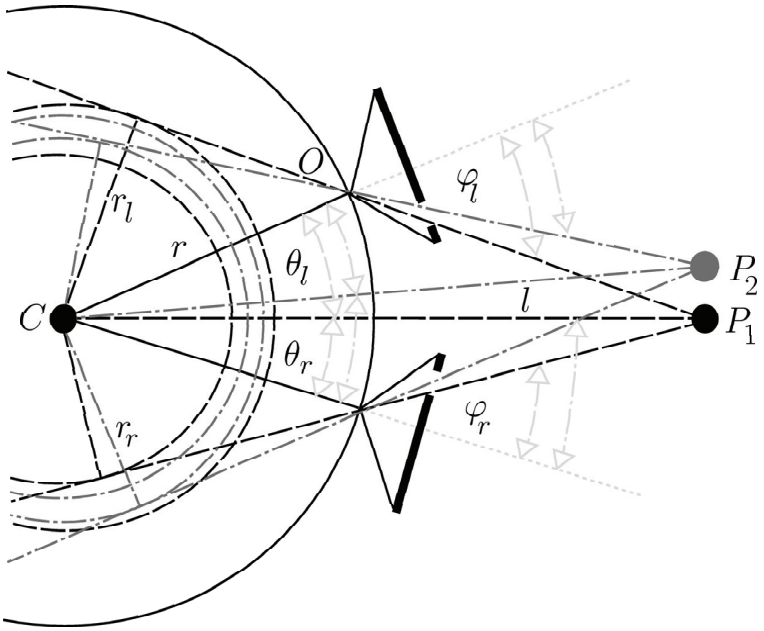

Figure 4: Angles θ_l and θ_r are related to angles φ_l and φ_r as presented in Eq. (4). Here the relationship is illustrated for two scene points.

We know that bigger φ brings bigger accuracy of the reconstruction process (see Sec. 3). And since we would like to achieve the best accuracy possible, we take bigger φ from the two possible values (φ_l and φ_r) and associated θ and calculate the depth estimation using Eq. (1).

Note that the vertical reconstruction is addressed in the same way as in the basic system [13].

The same reconstruction principle could also be used when we do the reconstruction from the non-symmetric pair of panoramas build from only one pixel column of the captured images. The only difference in this case is that φ_l and φ_r take only the value given by the pixel column that contributes to each panoramic image. The experiment in Sec. 8.1 proves the correctness of the principle.

7 Epipolar constraint

In the previous section we assumed that we can calculate the angle θ using Eq. (2). This equation holds if we do the reconstruction based on a symmetric pair of stereo panoramic images, which are made from one pixel column of the captured image. In this case we know that the epipolar lines are corresponding rows of the panoramic image (Sec. 3).

The stripe is composed of columns, each of them with a different angle φ. This basically means that we are dealing in fact with non-symmetric cases, for which the epipolar lines are different from corresponding rows. But if we look at the situation

from another viewpoint, we can establish the following: We are using symmetric stripes to build a stereo pair of panoramic images. If we lower the resolution of the captured image, we transform the stripe into a column. The symmetric stripes would become symmetric columns and we could again use the rows of the panoramic image as epipolar lines. The same conclusion can be drawn from the property of the viewing circle, which gets thicker if we use a stripe instead of a column.

We can see this fact already by observing the light rays forming the column or the stripe (Fig. 3) that contributes to the panoramic image. In both examples the viewing circle should be thicker as presented, because the column and the stripe are formed from the set of light rays, which intersect in the optical center of the camera.

Lower resolution also brings considerable decrease in the number of possible depth estimates [13], so the quality of results obtained from stripes should be better than the quality of results obtained from columns captured at a suitable lower resolution. For illustration: from around 140 possible estimates for *camera #1* in Sec. 8 to only around 10 estimates for the same camera at 14 times lower resolution – the width of the stripe that is transformed into a column W_s=14. It is also much harder to find corresponding points in such a low resolution image.

One very interesting property of the system is also that there is a possibility of reconstructing the scene from non-symmetric pairs of panoramas while still using the simple, horizontal epipolar constraint (see the experiment in Sec. 8.1).

8 Experimental results

In the experiments the following cameras were used (the width W of the captured images is 160 pixels in all experiments):

- *camera #1* with parameters: $\alpha = 34°$, $\beta = 25°$, $r = 30$ cm, $2\varphi_{\max} = 29.75°$, $2\varphi_{\min} = 2\varphi_{\max} - 2(W_s - 1)\frac{\alpha}{W}$ (note that $2\varphi = 29.9625°$ has been used when $W_s = 1$), $\theta_0 = W_s \cdot 0.205714°$,
- *camera #2* with parameters: $\alpha = 39.72°$, $\beta = 30.54°$, $r = 31$ cm, $2\varphi_{\max} = 34.755°$, $2\varphi_{\min} = 2\varphi_{\max} - 2(W_s - 1)\frac{\alpha}{W}$, $\theta_0 = W_s \cdot 0.257143°$,
- *camera #3* with parameters: $\alpha = 16.53°$, $\beta = 12.55°$, $r = 35.6$ cm, $2\varphi_{\max} = 14.46375°$, $2\varphi_{\min} = 2\varphi_{\max} - 2(W_s - 1)\frac{\alpha}{W}$, $\theta_0 = W_s \cdot 0.102857°$.

These values ensure similar number of possible depth estimates (Sec. 3).

The normalized error of the estimated depth l in comparison to the actual distance d (in % of d) for the scene point i is given as:

$$ERR_{\%,i} = \frac{|l_i - d_i|}{d_i} \cdot 100.$$

Furthermore, the average error $AVG_{\%}$ (arithmetic mean) over n scene points is calculated. The second measure, which is in the results written right beside the first

one ($AVG_{\%}$), is the standard deviation, which reveals how tightly all the various estimated depths are clustered around the average error in the set of data.

Correspondences for each feature point on the scene used in the evaluation have been determined with a *normalized correlation* procedure [6] and rechecked manually for consistency.

The real time sensor has been simulated by rotating one standard camera for the angle determined by the width of the stripe W_s. Since our final goal is to determine the usability of our system for mobile robot navigation, we have performed all the tests on real world images, so that the results reflect the applicability of implemented ideas in the real world.

8.1 Reconstruction from non-symmetric pairs of panoramas

Experiment background: In Sec. 6 we have introduced the principle for reconstruction when we use symmetric stripes instead of symmetric columns. We have mentioned that the same principle could also be used when we do the reconstruction from the non-symmetric pairs of panoramas build from only one pixel column ($W_s = 1$) of the captured images.

And in Sec. 7 we have stated that we can still use the simple, horizontal epipolar constraint in some cases. These cases have been investigated by Shum and Szeliski [21] and they have concluded that even in the non-symmetric cases the epipolar constraint is sufficiently close to the horizontal epipolar constraint if either r/l or φ are small. They have even presented the approximate numbers: $r/l \leq 0.\overline{6}$, $2\varphi \leq 30°$. In the example given here, we satisfied both criteria.

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

8.2 Reconstruction from stripe panoramas

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

The results were obtained with *camera #1.*
Results: The results for four different widths of the stripes (W_s) are given in Tab. 2. Note that the width of the captured images from which the panoramic images have been constructed is 160 pixels.
Conclusion: As expected, the accuracy deteriorates with wider stripes. Though, if we compare the results obtained with W_s=2 and W_s=6 with the results obtained

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

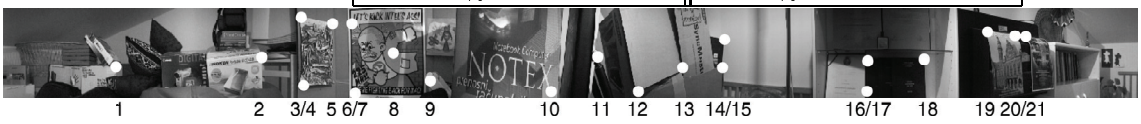

Table 1: The results obtained by processing symmetric pair of panoramas ($2\varphi = 29.9625°$) and non-symmetric pair of panoramas ($2\varphi_l = 29.9625°$, $2\varphi_r = 3.6125°$).

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

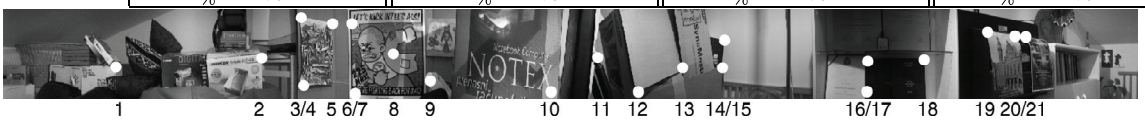

Table 2: The results obtained with four different widths of the stripes (W_s).

with W_s=1 (e.g. Tab. 1), we see that the results are still very good if not very similar.

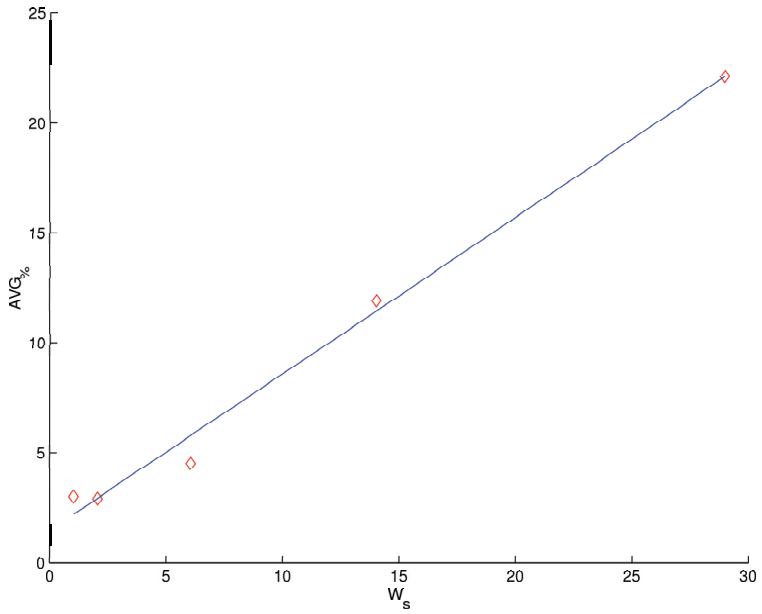

Figure 5: The relation between AVG$_\%$ and W_s (AVG$_\%$(W_s)) can be well approximated with a linear function. The diamonds represent the data involved in the approximation.

The graph in Fig. 5 shows that the relation between AVG$_\%$ and W_s (AVG$_\%$(W_s)) can be well approximated with a linear function, although we have fitted a second-order polynomial on the (sparse) data.

Another fact is that with the wider stripe more scene points are not captured in the panoramic images (Sec. 4.1), which means that the number of pixels in the left eye panoramic image that do not have the corresponding point in the right eye panoramic image increases. So, the generated panoramic images differ more and more from the ideal case (W_s=1) as we are increasing the width of the stripe W_s. In the ideal case each scene point in the left eye panoramic image has its corresponding point in the right eye panoramic image; exceptions are of course related to occlusions. – We performed an experiment in which we have used the *camera #1* (Sec. 8, W_s=1) to capture a banner written all over with text of different sizes, styles and fonts positioned at approximately 2 cm from the camera lens (on each location of the camera on the camera path). We were unable to prove from the human perceptual point of view the hypothesis that some scene points in this case are not captured. Really small deviations of the generated panoramic images from the real banner could as well be attributed to the fact that the images are discrete (for example: we cannot take a half of a pixel). On the other hand, the hypothesis is confirmed if stripes ($W_s > 1$) are used (Fig. 2).

8.3 Reconstruction from stripe panoramas — Different room

Experiment background: We want to see if we can achieve similar results as in Sec. 8.2, using the same camera (*camera #1*) in a different room?
Results: The results obtained for two most interesting stripe widths (according to Tab. 2) in the different room are presented in Tab. 3.
Conclusion: The overall results are a bit different, especially the results for W_s=14, though still similar. But this was also expected since the results for W_s=1 ($\text{AVG}_{\%}$=4.5% $\pm$ 4.5%) are also a bit worse compared to the results obtained in a previous room (the symmetric pair in Tab. 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. 3).

8.4 Reconstruction from stripe panoramas — Different cameras

Experiment background: We want to see how is the reconstruction error related to different cameras used in the same room, when the width of the stripe is constant for all cameras? As written in Sec. 8 the results were obtained at similar number of possible depth estimates.
Results: The comparison of results for the width of the stripe W_s=14 for three different cameras is given in Tab. 4. Note that for features marked 3, 5, 6, 7, 15, 19, 20 and 21 the real distance d in case of *camera #3* is different from the presented one. The reason for this lies in the vertical view angle of the camera β, which is smaller in comparison to other two cameras. This means that some marked feature points are not seen in the panoramic images generated with the *camera #3*, so we have chosen a nearby features with similar distances. By all means, in the calculations we have used the correct distances.
Conclusion: The results show that with bigger horizontal view angle α and at constant width of the stripe W_s (> 1) the reconstruction accuracy deteriorates:

$$\begin{gathered}\alpha_{camera\#3} < \alpha_{camera\#1} < \alpha_{camera\#2} \\ \Longleftrightarrow \\ \text{AVG}_{\%_{camera\#3}} < \text{AVG}_{\%_{camera\#1}} < \text{AVG}_{\%_{camera\#2}}.\end{gathered}$$

The result in Tab. 3 for W_s=14, gained with *camera #1*, but in the different room, also satisfies this relation.

Bigger horizontal view angle α means that the lens distortion in captured images is more obvious. Thus, this fact in relation with the above conclusion suggests that by using undistorted images, we could obtain better reconstruction accuracy. But the overall reconstruction accuracy obtained using undistorted sequence is very similar to that obtained using distorted sequence. This is due to the fact that we

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

Table 3: The results obtained in the different room, but with the same camera as in Sec. 8.2.

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

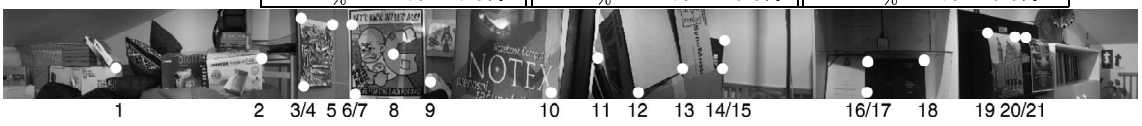

Table 4: The results obtained with three different cameras, while the width of the stripe W_s=14.

can see that the problem with missing scene points in the panoramic images (Sec. 4.1) remains.

So, one of the reasons for the lower reconstruction accuracy is again the fact that there are parts of the scene which are not captured in the panoramic image. This means that the number of pixels in the left eye panoramic image that do not have the corresponding point in the right eye panoramic image increases with bigger horizontal view angle α at constant width of the stripe W_s, while the shift angle θ_0 between two successively positioned cameras is bigger for bigger α (Sec. 8).

In Sec. 3 we have concluded that similar overall accuracy can be achieved if we use different cameras, while the width of the stripe W_s=1. If we relate this conclusion to the main conclusion of this section (with bigger horizontal view angle α and at constant width of the stripe W_s (> 1) the reconstruction accuracy deteriorates), we can conclude that the approximation presented in Fig. 5 gets steeper with increasing horizontal view angle α.

9 Conclusions

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

The presented theory and results suggest that the new sensor could be used for real time capturing of panoramic depth images and consequently for autonomous navigation of a mobile robot in a room. But this statement is unfortunately highly related to the application of the system and its demanded reconstruction accuracy. If in Sec. 3 we could conclude that by all means the basic system can be used for robot localization and navigation in a room, we are limited with the number of constraints when it comes to the real time sensor. On the other hand, assumptions made in this chapter have proved to be correct and revealed some other interesting properties of the system.

The main conclusions made in this chapter are:

- Building panoramic images from wider stripes than only one-pixel column brings faster execution of the building process and worse quality of the generated panoramas (Sec. 4).

- The stripes that contribute to the panoramic image do not cover all the scene (Sec. 4.1). The wider is the stripe, more scene points are not captured in the panoramic image (the number of points without correspondences increases (Sec. 8.2)). By using smaller r the gaps are smaller, which means that with smaller r we cover more scene points. The same goes for different values of α. This holds also for one-pixel column stripes.

- If we want to capture the parallax effect, which is essential for the reconstruction, we have to accept the fact that not all the scene points are captured (Sec. 4.1).

- If we could have a camera on each position on the circular path, we could build the panoramic image in real time and achieve the same quality of results as in case of the basic system, but in practice we should make a compromise between the width of the stripes, the number of cameras in respect to their parameters (e.g. α and external dimensions), the value of radius r and the needed accuracy of the system (Sec. 5).
- The stereo reconstruction procedures along with the epipolar constraint have been proved correct for both investigated cases: the basic sensor and the real time sensor (Secs. 6, 7 and 8).
- Using images with resolution 160×120 represents a good compromise between overall time complexity of the system and its accuracy (Secs. 3 and 5).
- Even in some cases of non-symmetric pairs of panoramas we can use our reconstruction procedure (Sec. 8.1).
- The reconstruction accuracy deteriorates approximately linearly with wider stripes (Sec. 8.2).
- We can achieve similar reconstruction accuracy with panoramas build from stripes ($W_s > 1$) with fixed size W_s of the captured images in different rooms at fixed horizontal view angle α (Sec. 8.3), but with bigger horizontal view angle α and at constant width of the stripe W_s (> 1) the reconstruction accuracy deteriorates (Sec. 8.4).
- In general, by undistorting the image sequence and building panoramas from stripes ($W_s > 1$), we do not obtain better reconstruction accuracy (Sec. 8.4).
- The approximate linear dependency of accuracy on the width of the stripes gets steeper with increasing horizontal view angle α (Sec. 8.4).

All this is true for the cameras used in the chapter, while for really wide angle cameras some conclusions perhaps demand further investigation in direction presented by the conclusion.

Acknowledgment

This work was supported by the Ministry of Education, Science and Sport of Republic of Slovenia (programme Computer Vision).

References

[1] Bakstein H., Pajdla T.: Panoramic Mosaicing with a 180° Field of View Lens. Proc. *IEEE Workshop on Omnidirectional Vision*, Copenhagen, Denmark, June, 2002, 60–67.

[2] Bakstein H., Pajdla T.: Ray space volume of omnidirectional 180°×360° images. Proc. *Computer Vision Winter Workshop (CVWW)*, Valtice, Czech Republic, February 3–6, 2003, 39–44.

[3] Benosman R., Kang S. B. (Eds.): *Panoramic Vision: Sensors, Theory and Applications*. Springer-Verlag, New York, USA, 2001.

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

[5] Chen S.: Quicktime VR — an image-based approach to virtual environment navigation. Proc. *ACM SIGGRAPH*, Los Angeles, USA, August 6–11, 1995, 29–38.

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

[7] 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, 2003, 48–56.

[8] Gupta R., Hartley R. I.: Linear pushbroom cameras. *IEEE Trans. PAMI*, 19(9), 963–975, 1997.

[9] 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, 2000. Available at: ftp://cmp.felk.cvut.cz/pub/cmp/articles/pajdla/Huang-TR-2000-07.ps.gz .

[10] Huang F., Wei S. K., Klette R.: Geometrical Fundamentals of Polycentric Panoramas. Proc. *IEEE ICCV*, Vancouver, Canada, July 9–12, 2001, I:560–565.

[11] Ishiguro H., Yamamoto M., Tsuji S.: Omni-directional stereo. *IEEE Trans. PAMI*, 14(2), 257–262, 1992.

[12] Peer P., Solina F.: Panoramic Depth Imaging: Single Standard Camera Approach. *International Journal of Computer Vision*, 47(1/2/3), 149–160, 2002.

[13] Peer P., Solina F.: Multiperspective panoramic depth imaging, chapter in *Computer Vision and Robotics*, John X. Liu, Ed. Nova Science Publishers, New York, pp. 135–188, 2006.

[14] Peer P., Solina F.: Where physically is the optical center? *Pattern recognition letters*, 27(10), pp. 1117–1121, 2006.

[15] Peleg S., Ben-Ezra M.: Stereo panorama with a single camera. Proc. *IEEE CVPR*, Fort Collins, USA, June 23–25, 1999, I:395–401.

[16] Peleg S., Pritch Y., Ben-Ezra M.: Cameras for stereo panoramic imaging. Proc. *IEEE CVPR*, Hilton Head Island, USA, June 13–15, 2000, I:208–214.

[17] Peleg S., Rousso B., Rav-Acha A., Zomet A.: Mosaicing on adaptive manifolds. *IEEE Trans. PAMI*, 22(10), 1144–1154, 2000.

[18] Peleg S., Ben-Ezra M., Pritch Y.: Omnistereo: Panoramic Stereo Imaging. *IEEE Trans. PAMI*, 23(3), 279–290, 2001.

[19] Rademacher P., Bishop G.: Multiple-center-of-projection images. *Computer Graphics (ACM SIGGRAPH)*, Orlando, USA, July 19–24, 1998, 199–206.

[20] 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, 2001, 42–47.

[21] Shum H. Y., Szeliski R.: Stereo Reconstruction from Multiperspective Panoramas. Proc. *IEEE ICCV*, Kerkyra, Greece, September 20–25, 1999, I:14–21.

[22] Shum H. Y., Kalai A., Seitz S. M.: Omnivergent Stereo. Proc. *IEEE ICCV*, Kerkyra, Greece, September 20–25, 1999, I:22–29.

[23] Sivic J.: Geometry of Concentric Multiperspective Panoramas. *M.Sc. Thesis*, Center for Machine Perception, Czech Technical University, Prague, Czech Republic, 2002.

[24] Sun C., Peleg S.: Fast Panoramic Stereo Matching Using Cylindrical Maximum Surfaces. *IEEE Trans. SMC – part B: Cybernetics*, 34(1), 760–765, 2004.

[25] Svoboda T., Pajdla T.: Epipolar Geometry for Central Catadioptric Cameras. *International Journal of Computer Vision*, 49(1), 23–37, 2002.

[26] 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, 2000, 289–294.

[27] Tanahashi H., Shimada D., Yamamoto K., Niwa Y.: Acquisition of Three-Dimensional Information in Real Environment By Using Stereo Omnidirectional System (SOS). Proc. *IEEE International Conference on 3-D Digital Imaging and Modeling*, Quebec City, Canada, May 28 – June 1, 2001, 365–371.

[28] 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, 1997, 243–250.

[29] Zomet A., Feldman D., Peleg S., Weinshall D.: Mosaicing New Views: The Crossed-Slits Projection. *IEEE Trans. PAMI*, 25(6), 741–754, 2003.