Towards a Real Time Panoramic Depth Sensor

Peter Peer and Franc Solina

University of Ljubljana
Faculty of Computer and Information Science
Tržaška 25, SI-1000 Ljubljana, Slovenia
{peter.peer,franc.solina}@fri.uni-lj.si

Abstract. Recently we have presented a system for panoramic depth imaging with a single standard camera. One of the problems of such a system is the fact that we cannot generate a stereo pair of images in real time. This paper presents a possible solution to this problem. Based on a new sensor setup simulations were performed to establish the quality of new results in comparison to results obtained with the old sensor setup. The goal of the paper is to reveal whether the new setup can be used for real time capturing of panoramic depth images and consequently for autonomous navigation of a mobile robot in a room.

1 Introduction

Real time panoramic depth imaging is an issue that is not well covered in the literature. There have been attempts or discussions [5] about it, but nothing has been done in practice so far, at least not by using mosaicing concept, i.e. the multiperspective panoramas.

In [6] we have presented a system for capturing panoramic depth images with a single standard camera. A stereo pair of images is captured while the camera rotates around the center of the system in a horizontal plane. The motion parallax effect which enables the reconstruction can be captured because of the offset of the cameras' optical center from the systems' rotational center. The camera is moving around the rotational center in angular steps corresponding to one vertical pixel-column of the captured standard image. A symmetrical 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 symmetrical column on the left side of the captured image contributes to the right eye panoramic image. This system however cannot generate panoramic stereo pair in real time. To illustrate this fact, we can write down the following example from practice: if the system builds a panoramic stereo pair from standard images with resolution of 160×120 pixels, using a camera with the horizontal view angle $\alpha = 34°$, it needs around 15 minutes to complete the task.

Generally mosaic-based procedures for building panoramic images [1,2,3] [5,6,7,8,9] can be marked as non-central (we are not dealing with only one center of projection), they do not execute in real time and they give high resolution results. Thus the procedures are not appropriate for capturing dynamic scenes.

N. Petkov and M.A. Westenberg (Eds.): CAIP 2003, LNCS 2756, pp. 107–115, 2003.

The main advantage of this procedures over other panoramic imaging systems (like catadioptric systems [10]) is the ability to generate high resolution results. But high resolution results are essential for effective depth recovery based on the stereo effect.

In the next section, the geometry of the system presented in [6] is revealed. In Sect. 3 we indicate how the time needed for the generation of a symmetric panoramic stereo pair can be dramatically reduced and in Sect. 4 we go even further and explain how we can achieve real time execution. Sect. 5 presents the depth reconstruction equation for the new setup. The epipolar constraint is discussed in Sect. 6. The evaluation of results is given in Sect. 7. We end the paper with conclusions in Sect. 8.

2 Geometry of the System

Let us begin this section with a description of the old sensor geometry [6].

The geometry of the system for creating multiperspective panoramic images is shown in Fig. 1. Panoramic images are then used as an 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 and the distance from point P to point O is denoted with d. θ is the angle between the line defined by point C and point O and the line defined by point C and point P. In the panoramic image the horizontal axis presents 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. With φ we denote 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 (this column contains the projection of the point Q), and the line defined by point O and the column of pixels that will be mosaiced in panoramic image (this 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 first idea about how to capture a stereo pair quicker is to generate panoramic images from wider vertical stripes instead of just one column.

3 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. But the real problem arises from the fact that we are not correcting the radial distortion of the cameras' lens. As will be shown in experimental results (Sect. 7), we were satisfied with

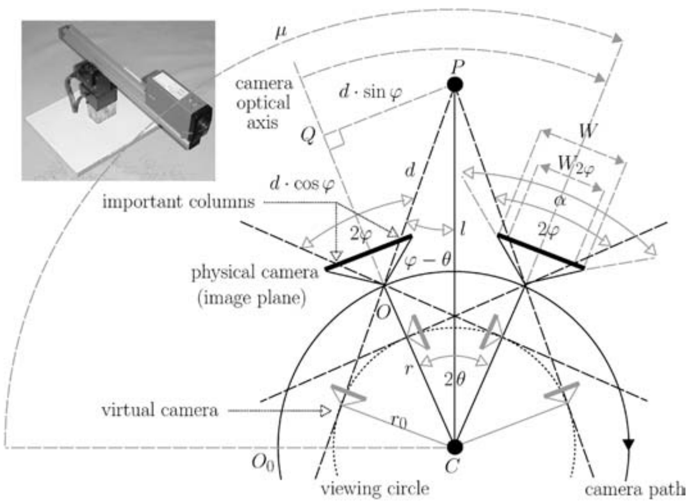

Fig. 1. Geometry of our system (old sensor setup) for constructing multiperspective panoramic images. Note that a ground-plan is presented; the viewing circle extends in 3D to the viewing cylinder. The optical axis of the camera is kept horizontal. In the small photograph the hardware part of the system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

the result when we used the stripe which was 14 columns wide and we think that it represents a good compromise. This statement is naturally highly related to the camera that we are using. The horizontal view angle of the camera is 34°, which means that 14 columns represents the angle of approximately 3°. In this case the building process takes around a minute.

3.1 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 radial 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). Of course this is partly because of the fact that we are dealing with the images, which are discrete (for an instance: we cannot take a half of a pixel), but if we take a look at the geometry of the system, 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.

We can write one more conclusion in regard to this property. Namely, the wider is the stripe, more scene points are not captured in the panoramic image (Fig. 2). And this holds regardless of the position of the stripe in the captured image. Naturally, by using columns we achieve best possible result (Fig. 2), though still not perfect, since the described property still holds, but is not so obvious.

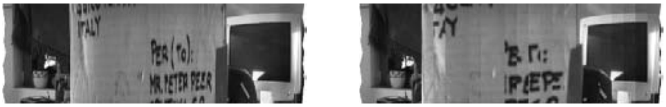

Fig. 2. The wider is the stripe, more distant are the scene points from the center of rotation that are not captured in the panoramic image: the left panoramic image was built from columns, while the right panoramic image was built from stripes (stripe was 14 columns wide). Note how very distant scene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

4 Achieving Real Time

The idea for a real time panoramic sensor is actually very simple. In our old system [6] 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 resolution of 160×120 pixels, then we have to put the cameras with the horizontal view angle $\alpha = 34°$ approximately $0.2°$ apart from each other and we need 360/0.2=1800 cameras.

In the case when we use stripes, the presented numbers get more reasonable. A 14 column stripes suggest that the cameras would be approximately $3°$ apart from each other and we would need 120 cameras to cover the whole circular path. If we use a camera with a wider horizontal view angle (e.g. $\alpha = 90°$), we need less cameras (e.g. 45). The new sensor does not need any moving parts, which means that we are not dealing 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.

5 Reconstruction

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

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

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

But when we use the 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

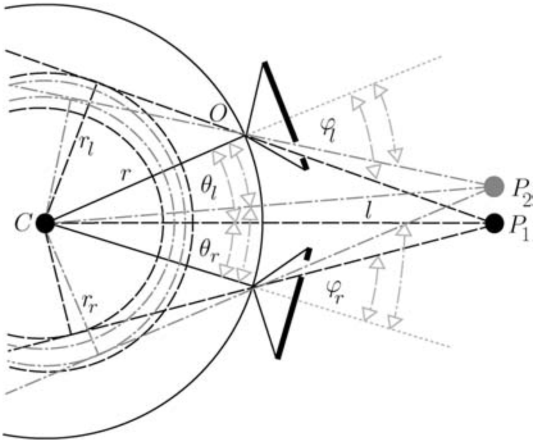

Fig. 3. Angles θ_l and θ_r are related to angles φ_l and φ_r as presented in Eq. 4. Here the relationship is illustrated for two scene points

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 [6] (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 3):

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

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. 3). 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:

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

We know that bigger φ brings bigger accuracy of the reconstruction process [6]. 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).

6 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 [6,9].

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 view point, 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.

7 Experimental Results

Fig. 4 shows some results of our new system. On the bottom image an example of the left eye stereo panoramic image is given. Symmetric stereo panoramic pair was build from stripes determined by $2\varphi_{\max} = 29.75°$ and $2\varphi_{\min} = 24.225°$. The stripe was 14 columns wide. The whole process was simulated (by rotating one standard camera) using radius $r = 30$ cm and the camera with the horizontal view angle $\alpha = 34°$.

For this image a sparse range image was calculated, which is presented in the middle image in Fig. 4. The sparse depth image was build by first detecting vertical edges in panoramic images. This information is normally essential for robot navigation. Edges were derived by filtering the panoramic images with the Sobel filter for searching the vertical edges [4,5]. We searched only for the correspondences of these feature points on the input panoramic images. All results were generated by using normalized correlation technique [4] with a correlation window of size $(2n+1) \times (2n+1)$, $n = 4$. We searched for corresponding points only on the panoramic image row which was determined by the epipolar geometry. We also used back-correlation procedure [4] and the information about the confidence in estimated depth [4], which we get from the normalized correlation estimations. In this way we increase the confidence in estimated depths. Black

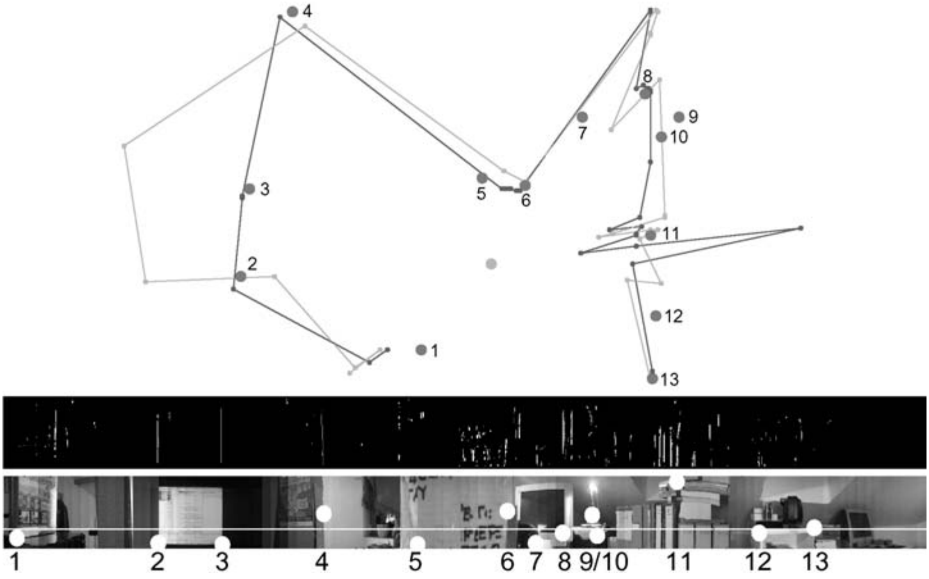

Fig. 4. The top image is a ground-plan showing the results of the reconstruction process based on the 85th row of the depth image (the middle image) from stereo pair built from stripes (*the lighter outline*) and only one column (*the darker outline*) of the captured images. The bottom image shows the reconstructed row and the features on the scene for which we measured the actual depth by hand. For orientation, the distance to the dot marked 1 is 63.2 cm

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.

Since it is hard to evaluate the quality of generated depth images, we present a reconstruction of the room from the generated depth image on the top image in Fig. 4. Now we are able to evaluate the quality of the generated depth image and consequently the quality of the system.

The result of the (3D) reconstruction process is a ground-plan of the scene. The following properties can be observed in Fig. 4: Big dark dots denote features on the scene for which we measured the actual depth by hand. A big light dot near the center of the reconstruction shows the center of our system. Small dots are reconstructed 3D points on the scene. Lines between small dots denote links between two successively reconstructed 3D points. The darker small points and lines were gained from panoramic images built from only one column of each captured image ($2\varphi = 29.9625^o$). The lighter small points and lines were gained from panoramic images built from stripes. The result shows the reconstruction based on the 85th horizontal row of the depth image. Small dots are reconstructed on the basis of estimated depth values, which are stored in the same row of the depth image. Note that the features in the scene marked with big dark dots are not necessarily visible in the same row.

Based on this reconstruction we can see that the darker outline has one problem on the right side, while the lighter outline has one on the left side. Besides that the reconstruction is pretty consistent. Generally speaking the darker outline is better then the lighter outline. This was expected, since the nature of the panorama building process says that the quality of the depth estimations is better when φ is bigger [6]. At the same time, the quality of the lighter outline is much better than the outline which would result from the panoramic image build from one column of each captured image at a suitable lower resolution. As already mentioned, this lower resolution turns stripes into columns. But lower resolution also brings considerable decrease in the number of possible depth estimates [6]: in this presented case from around 140 possible estimates to only around 10 estimates.

At the end let us present one quantitative measure, which gives the average error of the estimated depth (l) in comparison to the actual distance (d) over 19 scene points: $AVG_{columns} = ((\sum_{i=1}^{19} |l_i - d_i|/d_i)/19) \cdot 100\% = 4.3\%$, $AVG_{stripes} = 16.1\%$. In the last result ($AVG_{stripes}$) three points were really critical, while without them the result would be much better: 5.4%.

8 Conclusions

The presented theory and initial 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. Assumptions made have proved to be correct and revealed some other interesting properties of the system.

Since we can trust in estimates that are not far away from the center of rotation and the size of the angle φ prescribes the number of possible depth estimates [6], stripes suggest to dynamically modify the level of trust, since the angle φ within the stripe varies, while the procedure based on one column has a fixed angle φ. (See the two most left big dark dots and their reconstructions in Fig. 4.) We are also interested in how the system would perform in practice if we use a wide-angle camera [1] and if we correct the distortions in the captured images. This will be the subject of our future work.

References

1. Bakstein, H., Pajdla, T.: Panoramic Mosaicing with a 180° Field of View Lens. Proc. IEEE Workshop on Omnidirectional Vision. Copenhagen, Denmark (2002) 60–67
2. Benosman, R., Kang, S. B. (eds.): Panoramic Vision: Sensors, Theory and Applications. Springer-Verlag, New York, USA (2001)
3. Chen, S.: Quicktime VR — an image-based approach to virtual environment navigation. Proc. ACM SIGGRAPH. Los Angeles, USA (1995) 29–38
4. Faugeras, O.: Three-Dimensional Computer Vision: A Geometric Viewpoint. MIT Press, Cambridge, Massachusetts, London, England (1993)
5. Ishiguro, H., Yamamoto, M., Tsuji, S.: Omni-directional stereo. IEEE Trans. PAMI **14(2)** (1992) 257–262

6. Peer, P., Solina, F.: Panoramic Depth Imaging: Single Standard Camera Approach. Int. J. Comp. Vis. **47(1/2/3)** (2002) 149–160
7. Peleg, S., Rousso, B., Rav-Acha, A., Zomet, A.: Mosaicing on adaptive manifolds. IEEE Trans. PAMI **22(10)** (2000) 1144–1154
8. Peleg, S., Ben-Ezra, M., Pritch, Y.: Omnistereo: Panoramic Stereo Imaging. IEEE Trans. PAMI **23(3)** (2001) 279–290
9. Shum, H.Y., Szeliski, R.: Stereo reconstruction from multiperspective panoramas. Proc. IEEE ICCV, Vol. I. Kerkyra, Greece (1999) 14–21
10. Svoboda, T., Pajdla, T.: Epipolar Geometry for Central Catadioptric Cameras. Int. J. Comp. Vis. **49(1)** (2002) 23–37