Rendering novel views from a set of omnidirectional mosaic images [*]

Hynek Bakstein Tomáš Pajdla

Center for Machine Perception, Dept. of Cybernetics
Faculty of Electrical Eng., Czech Technical University
121 35 Prague, Czech Republic
{bakstein,pajdla}@cmp.felk.cvut.cz

Abstract

We present an approach to rendering stereo pairs of views from a set of omnidirectional mosaic images allowing arbitrary viewing direction and vergence angle of two eyes of a viewer. Moreover, we allow the viewer to move his head aside to see behind occluding objects. We propose a representation of the scene in a set of omnidirectional mosaic images composed from a sequence of images acquired by an omnidirectional camera equipped with a lens with a field of view of 183°. *The proposed representation allows fast access to high resolution mosaic images and efficient representation in the memory. The proposed method can be applied in a representation of a real scene, where the viewer is supposed to stand at one spot and look around.*

1. Introduction

Suppose we want to create a virtual walk through a real scene. In general, we have two options. Either we can create a 3D model of the scene and render novel views using that model and some rendering technique, or we can employ an image based rendering approach, which means that we render novel views of a scene using a sequence of images without any model of the scene. We sample light rays from the images and compose them into a novel image.

The first approach has the advantage that we are not limited by the viewpoint position, we can change the lighting, augment a virtual object into the scene, etc. But we have to recover an underlying 3D model of the scene. This might be suitable for some simple scenes, but most of the real scenes are too complex for this to be feasible.

The second approach (referred to as *image based rendering*, *IBR*) is therefore commonly used for virtual walkthroughs in real scenes [6, 15, 14]. The main drawback of image based rendering is the tradeoff between the freedom of movement and the amount of data required to represent a scene. It is possible to capture a complete plenoptic function [1], however, since this function is five dimensional (if we represent light rays by a point in the space and a direction and ignore the time and the wavelength), the resulting amount of data is huge. Several approaches to reducing the dimensionality of the plenoptic function were proposed [7, 8, 5, 4, 15, 14]. The reduction of the dimensionality results in some restriction on the freedom of movement of the viewer. For example in [6], only a panoramic image (2D parameterization of the plenoptic function) is used and therefore the viewer is only allowed to rotate his head.

We adopt a representation of the scene using the plenoptic function reduced into 3D. The reduced plenoptic function is called *the ray space volume* in this paper. Main contribution of this paper is that it combines together images from an omnidirectional camera into a ray space volume consisting of high resolution omnidirectional mosaic images covering $360^\circ \times 180^\circ$ field of view. There are approaches using omnidirectional images for image based rendering [13, 16]. Our method differs from them in generating omnidirectional mosaics apart from having omnidirectional images as the input for the mosaic composition. This allows rendering views from mosaics when looking in all directions in the scene, see Figure 1. We are also able to compose a stereo pair for arbitrary viewing direction because our camera was moved on a circular path while acquiring of input images, unlike in [13, 16], where the camera was moving on a line, and therefore no stereo views could be rendered in the direction of movement. We also discuss the topic of fast access to data and efficient representation in memory because high resolution of mosaics results in a huge amount of data. We show how to create a stereo pair of virtual views looking in arbitrary direction and how to simulate a sideways motion of the viewer to see behind the occluding objects.

In Section 2 we describe the creation of volumes and their properties. In Section 3 we show how to create om-

---

[*] This work was supported by the following grants: GACR 102/01/0971, BeNoGo IST-2001-39184, MSMT Kontakt 22-2003-04, and MSM 212300013. We would also like to thank to Václav Větvička, the manager of the Botanical garden of the Charles University, for cooperation.

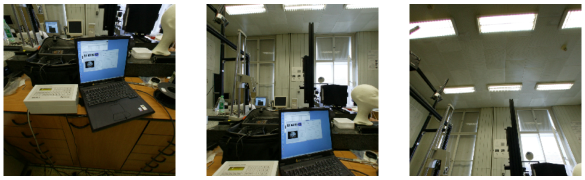


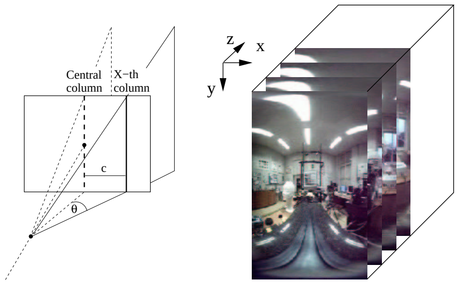
Figure 1: Synthesized images for a viewer looking down, forward and up. All views were synthesized from one non-central mosaic image. Images by courtesy of Branislav Mičušík.

nidirectional concentric mosaics. Section 4 describes the concept of so called viewing windows, and Section 5 is devoted to the description of efficient data representation of a ray space volume with respect to easy access to novel views. Experimental results are listed in Section 6.

2. Ray space volumes

In this section we briefly describe the construction of ray space volumes of images from a sequence captured by a rotating camera. We choose rotation of the camera instead of translation in order to acquire panoramas covering the whole $360°$ around the viewer. We focus on the composition of the volume from omnidirectional images and on the composition of $360° \times 180°$ mosaic images from these volumes.

By *volume* we mean images stacked into a 3D matrix, as it was proposed in [11]. Ideally, we work with images, where columns correspond to light rays lying in one plane that contains the optical center of the camera and is parallel to the rotation axis, see Figure 2(a). However, since the optics of the camera may introduce nonlinear distortion and the camera can be tilted, all these phenomena have to be compensated for so that the correspondence between angles and columns holds. In the following text we suppose that the images in the sequence are calibrated [2, 9] and that columns really correspond to the angle θ as it is depicted in Figure 2(a).



(a) (b)

Figure 2: (a) Angle θ corresponds to the column c in the image. (b) Ray space volume composed of stacked images.

Ray space volumes consist of images stacked into one huge three dimensional matrix so that each image lies in a plane parallel with the xy plane. The images are stacked along the z axis, see Figure 2(b). Sometimes, the z axis corresponds to time. In our case, the z axis determines the rotational position in which the image was acquired, hence the term ray space volume instead of space time volume in [15]. The x and y axis determine the image column and row respectively. The x axis also corresponds to the angle θ between a plane containing the central column with the camera center and the plane containing the x-th column and the camera center, see Figure 2(a). Note that the central column contains the image center, but it does not have to be the middle column of the image.

If we cut the volume by a plane parallel to the yz plane, that means that the x value is fixed, we get a mosaic composed of x-th columns of each image as it is depicted in Figure 3. By changing the value of x, we can create different mosaics corresponding to different angles.

We can also cut the volume by a plane parallel to the xz plane. We do not obtain an epi image [3] except for the cut by the plane containing the circle of rotation, since there is in general no epipolar plane except for the plane containing the circle of rotation. Instead, we get a planar cut through epipolar hyperboloids. Each row of the image corresponds to one row in a mosaic image. This row corresponds to one ruling in a stereo correspondence surface, in this case a single sheeted hyperboloid [10].

If we want to compose the volume from omnidirectional images, we have to calibrate the omnidirectional sensor [2, 9] in order to create an image, where columns correspond to light rays in a plane holding certain angle θ with the optical axis, as it is depicted in Figure 2(a). We can then rearrange pixels corresponding to the rays into a column and stack these columns into the volume images, see Figure 4. Then we can access the volume in the same way as described above for classical cameras. We use a lens instead of a mirror in our realization of the omnidirectional sensor, because a mirror always obscures part of images. The lens gives us an unobscured image in the whole field of view.

3. Omnidirectional concentric mosaics

Concentric mosaics [12] can be created from a sequence of images captured by a perspective camera rotating on a circular path with radius R. From each image in the sequence, we select one column (the same for all images) and paste it into the mosaic. The resulting mosaic image can be considered as an image captured by some mosaic camera, which does not have to have a single center of projection. Instead, the light rays are tangent to a cylinder and they all intersect the circle of rotation. Recall Figure 2(a), where each col-

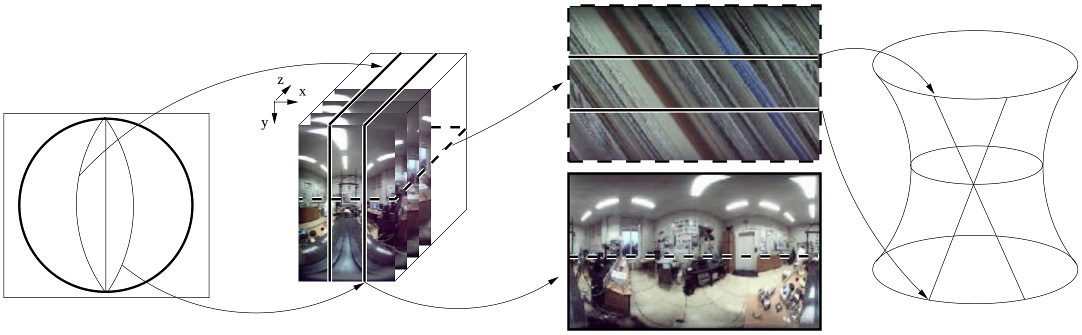


Figure 3: A cut through a volume parallel to the yz plane results in a mosaic image. By cutting the volume by a plane parallel to the xz plane we obtain en EPI like image. Symmetric concentric mosaics do not have epipolar geometry consisting of planes, but of a single sheet hyperboloids.

umn in the image corresponds to some angle θ. The value of θ determines the radius $r = R\sin(\theta)$ of the cylinder to which are the light rays tangent, see Figure 5. If θ equals $0°$, all the light rays intersect the axis of rotation. In case of θ equals $90°$, the light rays are tangent to the circle of rotation.

The angle θ corresponds to the x position of the plane parallel to the plane yz by which we cut the volume. Since we can create the volume from omnidirectional images, we can also compose omnidirectional concentric mosaics from the volume in a straightforward way. The resulting omnidirectional concentric mosaics then can cover $360° \times 180°$ field of view, thus capturing all light rays emanating from the circle of rotation. The respective noncentral camera creating the mosaics then observes everything except the inner part of the cylinder created by lifting the viewing circle in the direction of the axis of rotation.

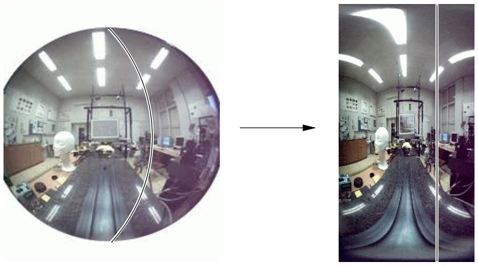


Figure 4: Omnidirectional images have to be preprocessed before creating a volume.

4. Viewing window

One possible application of IBR is displaying stereo pairs of images to a viewer wearing a head mounted display (HMD). HMDs have two small displays with a certain limited field of view. Therefore we want to synthesize two views from two mosaics, one for each eye, covering a smaller part of the scene. We realize that by a so called *viewing window* which

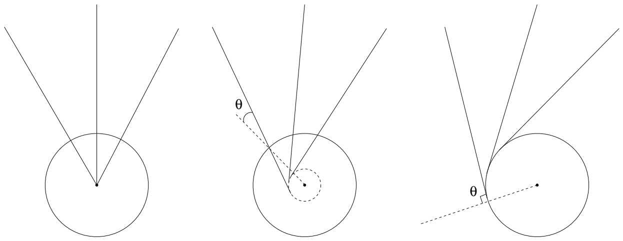


Figure 5: Values of θ angle of the light rays determine radii of circles to which are the light rays tangent. For $\theta = 0°$, the light rays intersect the axis of rotation, for $\theta = 90°$, they are tangent to the circle of rotation.

can be slid on the mosaic images to simulate change in the viewing direction, see Figure 6. Therefore, if the viewer is not allowed to change his position but just to rotate his head, we need only two mosaic images to represent the scene. Since our mosaic images are omnidirectional, the viewer can also turn his head up and down.

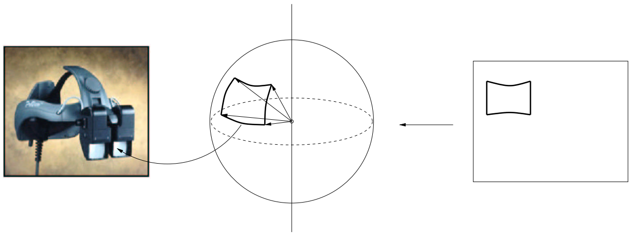


Figure 6: Viewing window simulates the limited field of view of the eyes of the viewer. Image of the HMD is from Kaiser Electro-Optics, Inc.

The field of view and the number of pixels in both directions of the viewing window determine the number of pixels of the two mosaics. If the viewing window covers φ_v degrees in the vertical direction and φ_h in the horizontal direction with respective number of pixels r_v and r_h, then the mosaics should have

$$mr_v = \frac{180^\circ}{\varphi_v} r_v, mr_h = \frac{360^\circ}{\varphi_h} r_h \qquad (1)$$

pixels in the vertical and horizontal direction respectively.

The mosaics are composed so that we sample the light rays constantly and paste pixels from the images into the mosaic images so that a constant increment in angles results in a constant increment in pixels, in both directions. Due to this mosaic composition, depicted in Figure 7, objects in the mosaic images are significantly distorted. Prior to rendering images in the viewing window, we have to undistort them. We cannot undistort the whole mosaic since it can be mapped on a sphere but not on a plane. However, smaller parts of the image can be mapped on a plane. This can be done in a very efficient way by a rendering hardware.

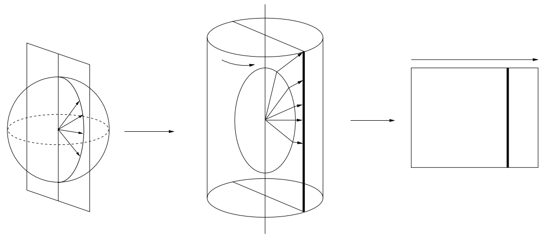
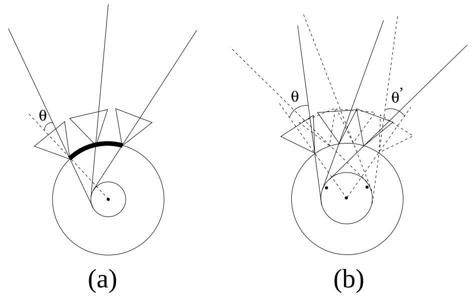


Figure 7: Light rays are mapped from a sphere to a cylinder and then the cylinder is unrolled onto a plane resulting in a mosaic image. Note that the bold curve and lines respectively denote light rays in one plane.

We map the mosaic on a sphere and then the viewing window is realized by a virtual camera placed in the center of the sphere, with fields of view identical to the simulated fields of view of the viewer, see Figure 6. The viewing direction of the viewer also corresponds to the viewing direction of the camera. We can use, (e. g.), OpenGL to do the mapping in real time using a hardware acceleration. Note that the viewing window contains only a subset of the columns of the mosaic image. These columns correspond to a subset of camera positions in the original sequence, see Figure 8(a). We can use this fact to reduce the number of images which have to be stored in the memory. This concept will be discussed later in this paper.

For a stereo sensation, we need to create two mosaics with disparity. We can realize that by creating two mosaics with different angle θ, as it is depicted in Figure 8(b). Note that as in the case of human eyes, which can be focused on close or distant objects, the disparity of the objects in the scene can be adjusted here by changing the vergence angle.

We also do not have to limit the location of the viewer to just a single point. Imagine that there is an object in the scene occluding a view of something interesting behind it. It is natural just to move one's head a little so that the occlusions change, without moving the whole body, i. e. without changing dramatically the position of the viewer. It has been shown that such a change in occlusion can be simulated by using a set of concentric mosaics [12, 14]. We investigate this approach in the next section focusing on a fast access to the respective views in the volume.



(a) (b)

Figure 8: (a) Only a subset of the images contributes to an image in a viewing window. (b) For a stereo sensation, we need to create two mosaics, each for different eye position.

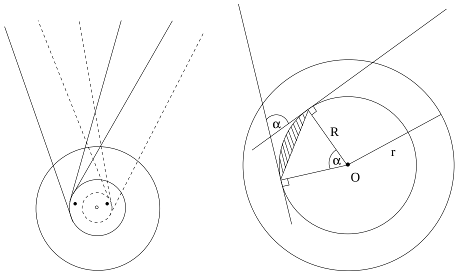
5. Data representation of ray space volumes

The images that the viewer can see by moving the viewing window are not central [10]. The light rays composing one mosaic are tangent to a cylinder. Light rays in the viewing window are tangent only to a section of the cylinder and the size of this section is determined by the horizontal FOV α of the 'eye' of the viewer, see Figure 9(b). From now on, we will refer to this section of the cylinder as to the eye of the viewer. Since the objects are typically in a distance from the center of rotation O that is much higher than the radius of the circle of rotation r, the stereo sensation is almost the same as from central images.

We have mentioned the *position of the viewer*. It represents a location of the two eyes for which we create virtual images. Those eyes have a field of view which determines the viewing window size. When the viewer is positioned in the center of the circle of rotation, the eyes have the same distance from this center, see Figure 8(b). Note that the light rays composing the images for both eyes are tangent to the same cylinder. If the viewer turns his head, the eyes are rotated, but the tangent cylinder does not change. If the viewer is moving sideways, the light rays start to be tangent to two different cylinders, one for each eye, as it is depicted in Figure 9(a).

There is a relation between the position of the viewer and the angle θ. By increasing the radius of the cylinder, to which the light rays are tangent, and thus changing the angle θ, we move the viewer closer to the circle of rotation. There is a clear limiting case, the maximal value of the angle θ.

In our case this value is 90°, so the limiting position is exactly the circle of rotation. Also note that the position of the viewer can be moved only sideways because the light rays are always tangent to some cylinder, see Figure 9(a). For a forward motion, we would need to employ X-slit cameras [15] or rendering from concentric mosaics [12].



(a) (b)

Figure 9: (a) Concentric mosaics for different radii of the tangent cylinder can be used to simulate different positions of the eyes of the viewer. (b) Even when considering the viewing window, the concentric mosaic does not produce a central image.

The main reason for using just the concentric mosaics and not rendering novel views from them using either approach proposed in [12] or from images as in [15] is the speed and memory requirements. We can simulate the movement of the head of the viewer sideways so that he can see behind occluding objects and focusing on close or distant objects just by selecting different pairs of concentric mosaics without any additional computations.

The proposed approach to the representation of real scenes is as follows. We capture a sequence of omnidirectional images by a camera rotated on a circular path. Then we create a ray space volume from the images. Next we process the volume to obtain a set of omnidirectional mosaic images, which can be still considered as a ray space volume. The viewer has some initial orientation and vergence which corresponds to a pair of omnidirectional concentric mosaics. We load into the memory not only this pair, but also several adjacent pairs to allow a change of the vergence angle and sideways movement. When the viewer moves his head in one direction, we load more mosaics from the volume while discarding the mosaics corresponding to the opposite direction of the movement. This allows us to have a set of high resolution mosaics, uncompressed and ready to map via hardware, in the memory.

The images can be accessed in a different way as well, making use of the fact, that the viewing window determines a small subset of images which contribute to the view presented to the viewer. We can load only this subset into the memory plus some adjacent images into a cache. Then we will not be able to create the whole mosaic, but only its part contributing to the viewing window. When the viewer turns his head, more images will be required to be loaded into the memory, while the sideways movement may be simulated from the images which are already in the memory or in the cache. This is an opposite situation to the case described in the previous paragraph. If the looking around is the preferred scenario, the first method should be used. Otherwise, the second method is better.

6. Experimental results

Our experimental setup consisted of a Nikon FC-E8 lens mounted on a Pixelink high resolution color camera.The camera was rotated on a closed circular path with a radius of $30cm$. We captured two types of sequences, one in our lab and the other one in a botanical garden. Images provided by the camera have resolution of 1200×1024 pixels and their number was chosen so that the composed mosaics had unit aspect ratio. In our case, we captured ≈ 2800 images in order to able to compose the mosaics from single image columns only. We can use smaller number of images and compose the mosaic not from columns, but from stripes of columns from volume images.

Figure 10 illustrates simulated sideways movement of the viewer. Note dramatic change in occlusions and in the pose of the head of the figurine. Figure 11 depicts different vergence angles, the while lines help to visualize the disparity. The top row contains a pair of virtual views, for the left and the right eye respectively, with eyes focused on distant objects. Note the disparity of the railing, which is too big and the viewer is not able to perceive depth. The bottom row shows views of the same part of the scene with different vergence angle. Note smaller disparity change for the railing in foreground. This effect can be used to simulate verging of the viewer's eyes on a closer or a distant object.

7. Conclusion

We have shown how to create ray space volumes from omnidirectional images and how to create omnidirectional concentric mosaics from these volumes. With the help of these mosaics we are able to simulate different vergence of eyes of a viewer and his sideways movement. We also proposed how to organize high resolution mosaics so that they can be easily accessed and stored in the memory.

References

[1] E. H. Adelson and J. Bergen. *The Plenoptic Function and the Elements of Early Vision*, pages 3–20. MIT Press, 1991.

[2] H. Bakstein and T. Pajdla. Panoramic mosaicing with a $\mathbf{180}^\circ$ field of view lens. In A. D. Williams, editor, *Proceedings of Omnivis2002*, pages 60–67, IEEE press, 2002.

Figure 10: Sideways movement of the viewer is simulated by creating different concentric mosaics. Note the change in occlusions and in the pose of the head.

Figure 11: Different disparities correspond to different vergence angles. The disparity of the railing in the top row is too big, the eyes are vergent on distant objects Note smaller disparity in the bottom row.

[3] R.C. Bolles, H.H. Baker, and D.H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *IJCV*, 1(1):7–56, 1987.

[4] J. Chai, S. B. Kang, and H.-Y. Shum. Rendering with non-uniform approximate concentric mosaics. In *Second Workshop on Structure from Multiple Images of Large Scale Environments, SMILE*, pages 94–108, LNCS, July 2000.

[5] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *Procceding of the SIGGRAPH 2000 conference*, volume 1, pages 307–318, July 2000.

[6] S. E. Chen. QuickTime VR — an image-based approach to virtual environment navigation. *Computer Graphics*, 29(Annual Conference Series):29–38, 1995.

[7] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Procceding of the SIGGRAPH'96 conference*, pages 43–54, August 1996.

[8] M. Levoy and P. Hanrahan. Light field rendering. In *Procceding of SIGGRAPH'96*, pages 31–42, August 1996.

[9] B. Mičušík and T. Pajdla. Estimation of Omnidirectional Camera Model from Epipolar Geometry. In *Proc. of the CVPR2003 conference*, to be published, 2003.

[10] T. Pajdla. Epipolar geometry of some non-classical cameras. In B Likar, editor, *Proceedings of Computer Vision Winter Workshop*, pages 223–233, February 2001.

[11] Y. Pritch, M. Ben-Ezra, and S. Peleg. Automatic disparity control in stereo panoramas (omnistereo). In *IEEE Workshop on Omnidirectional Vision (OMNIVIS'00), Hilton Head, South Carolina*, pages 54–61, IEEE Press, June 2000.

[12] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. In *Procceding of the SIGGRAPH'99 conference*, volume 1, pages 299–306, August 1999.

[13] T. Takahashi, H. Kawasaki, K. Ikeuchi, and M. Sakauchi. Arbitrary view position and direction rendering for large-scale scenes. In *Proc. of the CVPR2000 conference*, volume 2, pages 296–303, IEEE Press, 2000.

[14] L. Wang, H.-Y. Shum, and S. B. Kang. Object representation and rendering using inverse concentric mosaics. In D. Suter and A. Bab-Hadiashar, editors, *Proceedings of ACCV 2000*, volume 1, pages 150–155., January 2002.

[15] D. Weinshall, M.-S. Lee, T. Brodsky, M. Trajkovic, and D. Feldman. New view generation with a bi-centric camera. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proc. of the ECCV'02 Conference*, volume 1, pages 614–628, LNCS, May 2002.

[16] K. Yamaguchi, H. Takemura, K. Yamazawa, and N. Yokoya. Real-time generation and presentation of view-dependent binocular stereo images using a sequence of omnidirectional images. In *Proc. of the ICPR2000 conference*, volume 4, pages 589–593, IEEE Press, 2000.