CALIBRATION PROCEDURE FOR A 360×360 MOSAIC CAMERA

Hynek Bakstein and Tomáš Pajdla
Center for Machine Perception, Czech Technical University in Prague
{bakstein,pajdla}@cmp.felk.cvut.cz

KEY WORDS: Omnidirectional vision, fisheye lenses, mosaic images, image based rendering.

ABSTRACT

We present a model and a calibration procedure for a 360×360 mosaic camera. The camera is composed of an omnidirectional camera rotated on a circular path, therefore it can observe 360° in both the vertical and the horizontal direction. It has a simple model and a linear epipolar geometry, when carefully set up. We show how to recover defects in an assembly of the camera using only images of corresponding points. Finally, we overview some practical realizations of the 360×360 mosaic camera. Possible applications of this camera include representation of a real environment for a virtual reality and 3D reconstructions.

1 INTRODUCTION

We present a model and a calibration procedure for a 360×360 mosaic camera in this paper. This camera has a simple geometric model which allows us to acquire a 3D information of a scene from a stereo pair of mosaic images. The closest work related to our paper is (Huang et al., 2003). Our work differs in a fact, that we combine a simple geometrical model of a non-central camera with image based representation of the scene, instead of a range data in (Huang et al., 2003). We suggest using not only a line camera but a fisheye lens with a classical camera to obtain more information about the scene. The fisheye image are then used as an input for an image based rendering (IBR) method. The scene depth recovered from the mosaic images is used to greatly reduce the number of images required for photorealistic rendering of virtual views using IBR. We do not require any range data or a priori 3D model.

Figure 1: An image from a 360×360 camera.

We propose to use a so called 360×360 camera, introduced by (Nayar and Karmarkar, 2000), as a device for acquisition of such a representation. This camera can capture its complete surroundings in a single image. In addition, this camera has a very simple geometrical model and allows a metric reconstruction of scene features (Bakstein and Pajdla, 2001) and it has also optimal stereo sensing properties (Shum et al., 1999). In practice, the camera is composed of an omnidirectional camera rotated on a circular path. Only a subset of light rays, lying in a plane, is selected from an image acquired by the omnidirectional camera. A real setup is depicted in Figure 2.

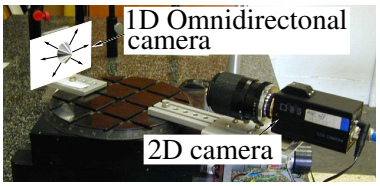


Figure 2: An example of a realization with a telecentric lens (2D camera) observing a conical mirror, reflecting light rays into parallel planes (1D omnidirectional camera). This camera-mirror rig is rotated on a circular path to create the 360×360 mosaic camera.

More formally, the 360×360 camera is composed of a 1D omnidirectional camera, which can be thought of as a planar pencil of light rays, rotated off-center on a circular path $\mathcal{C}$ in a plane δ, see Figure 3. Off-center rotation allows depth measurement. We present a complete geometrical analysis and calibration procedure for the 360×360 camera. Moreover, we suggest using a fisheye lens instead of mirrors employed in the original work for a practical realization of the 1D omnidirectional camera.

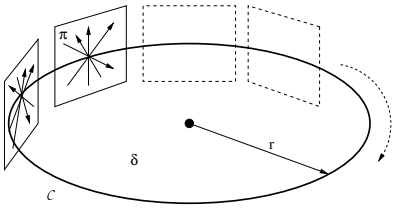


Figure 3: 360×360 mosaic camera is realized by a planar pencil π rotated on a circular path $\mathcal{C}$.

The rest of the paper is organized as follows. In Section 2, we describe a right symmetric 360×360 mosaic camera, the ideal geometry of the 360×360 mosaic camera, having the 1D omnidirectional camera perpendicular to the plane δ and tangent to the circle $\mathcal{C}$. We present its model, projection and image formation equations, and a procedure for

recovery of the model parameters. Then, in Section 3, we deal with issues introduced by a practical realization of the 360×360 mosaic camera, when the 1D omnidirectional camera is not perpendicular to δ and/or tangent to $\mathcal{C}$. A practical realization of the 1D omnidirectional camera is discussed in Section 4 and an application of the 360×360 mosaic camera in image based rendering is presented in Section 5.

2 RIGHT SYMMETRIC 360×360 MOSAIC CAMERA

In (Nayar and Karmarkar, 2000), it was assumed that the planar pencil of light rays is perpendicular to the plane of rotation δ and tangent to the circle $\mathcal{C}$. Then, we get a system which has a linear epipolar geometry. In fact, each point outside the circle $\mathcal{C}$ can be observed twice by the 360×360 camera. If we collect the first observations of each scene point in one image and the second observations in another image, we obtain a rectified stereo pair of images. However, this holds only when the assumptions about perpendicularity and tangency hold. We call this case a *right symmetric* 360×360 *mosaic camera* to distinguish it from the general case, which we refer to as 360×360 *mosaic camera*, discussed in Section 3. Here, we focus on the simple geometry. At first, we present equations for computation of a 3D point provided with image coordinates then we present projection equations.

Digital images are measured in pixels, but the mosaic camera model is described by angles. We have to define a relation between pixel coordinates (u, v) in the mosaic images and their corresponding angles α, α', and β. We define a coordinate frame in the mosaic images so that $u = 0$ when $\alpha = 0$ resp. $\alpha' = 0$ and $v = 0$ for $\beta = 0$, as it is depicted in Figure 4.

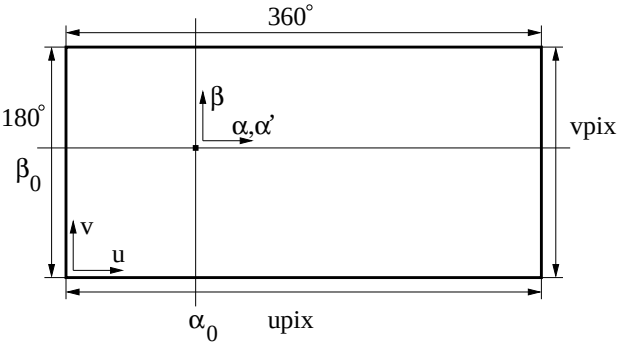


Figure 4: Mosaic image coordinate system (u, v) and its relation to the angles α, α', and β and their zero values α_0 and β_0.

Using basic trigonometric rules and identities, the following expressions for the x, y, and z can be derived:

$$x = r\frac{\cos\delta}{\cos\gamma} \quad (1)$$

$$y = r\frac{\sin\delta}{\cos\gamma} \quad (2)$$

$$z = r\tan\gamma\tan\beta \ , \quad (3)$$

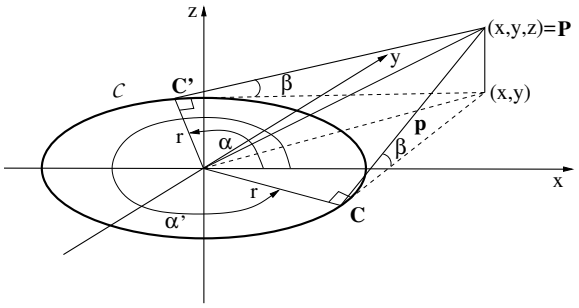


Figure 5: Right symmetric 360×360 mosaic camera geometry.

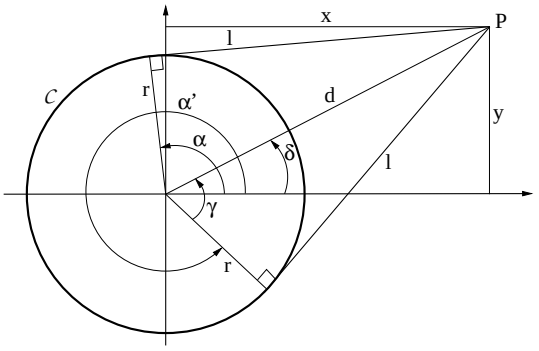


Figure 6: Right symmetric 360×360 mosaic camera geometry - a view in the $\mathbf{xy}$ plane.

where r is the radius of the viewing circle, $\gamma = \frac{\alpha - \alpha'}{2}$, and $\delta = \frac{\alpha + \alpha'}{2}$.

Alternatively, we can follow the approach of (Baratoff and Aloimonos, 1998). We can use a construction, where a scene point is reconstructed by intersecting a plane and a line in 3D. The line passes through the center of the 1D omnidirectional camera at position where it observes the scene point for the first time, the plane passes through this center at position, where the scene point is observed for the second time.

We use Plücker matrix L to represent the line and a homogeneous 4-vector $\boldsymbol{\pi}$ for the plane. Then, the scene point is computed as

$$\mathbf{X} = \mathrm{L}\boldsymbol{\pi} \ . \quad (4)$$

The line L and the plane $\boldsymbol{\pi}$ are computed as

$$\mathbf{C} = (r\cos\alpha, r\sin\alpha, 0) \, , \quad (5)$$

$$\mathbf{p} = (-\sin\alpha\cos\beta, \cos\alpha\cos\beta, \sin\beta) \, , \quad (6)$$

$$\mathrm{L} = \mathbf{C}(\mathbf{C}+\mathbf{p})^T - (\mathbf{C}+\mathbf{p})\mathbf{C}^T \, , \quad (7)$$

$$\boldsymbol{\pi} = (\cos\alpha', \sin\alpha', 0, -r) \, , \ . \quad (8)$$

To compute the angles α, α', and β from coordinates of a 3D point $\mathbf{P} = (x, y, z)$, we can write direct equations for α, α', and β:

$$l = \sqrt{x^2 + y^2} \, , \quad (9)$$

$$\gamma = \arccos\left(\frac{r}{l}\right) \, , \quad (10)$$

$$\delta = \arcsin\left(\frac{y}{l}\right) \, , \quad (11)$$

$$d = l \sin \gamma \,, \tag{12}$$
$$\alpha = \delta + \gamma \,, \tag{13}$$
$$\alpha' = 2\delta - \alpha \,, \tag{14}$$
$$\beta = \arctan\left(\frac{z}{d}\right) . \tag{15}$$

It can be seen in the above equations, that both the projection and reprojection equations depend on a single parameter, the radius of the circle $\mathcal{C}$. We will discuss recovery of its value in the following section.

2.1 Calibration from a known relative distance between two points

To compute the radius r, we only need to know a distance between two scene points $\mathbf{P_1}$ and $\mathbf{P_2}$. Nothing has to be known about their distance from the camera center $\mathbf{O}$. Recall the image formation by the right symmetric 360×360 mosaic camera (1)—(3). From the mosaic images, we can determine the angles γ_1, δ_1, γ_2, and δ_2 describing the light rays by which we observe the points $\mathbf{P_1}$ and $\mathbf{P_2}$. Since the reconstruction of the scene coordinates with unknown radius r differs from the correct one only by scaling, we can express their scene coordinates as a function of r and then compute r from a relation stating that the distance between these scene coordinates has to be exactly d. Without loss of generality, we can assume that the two points lie in the plane δ, that is, they both have $z = 0$, see Figure 7.

Let the scene coordinates of the point $\mathbf{P_1}$ be (compare with (1)—(3))

$$x_1 = r\frac{\cos \delta_1}{\cos \gamma_1} \,, \tag{16}$$
$$y_1 = r\frac{\sin \delta_1}{\cos \gamma_1} \,, \tag{17}$$
$$z_1 = 0 \tag{18}$$

and for the point $\mathbf{P_2}$

$$x_2 = r\frac{\cos \delta_2}{\cos \gamma_2} \,, \tag{19}$$
$$y_2 = r\frac{\sin \delta_2}{\cos \gamma_2} \,, \tag{20}$$
$$z_2 = 0 \,. \tag{21}$$

We abbreviate $\frac{\cos \delta_i}{\cos \gamma_i}$ by p_{xi} and $\frac{\sin \delta_i}{\cos \gamma_i}$ by p_{yi}. Then, the constraint that the distance between $\mathbf{P_1}$ and $\mathbf{P_2}$ is d

$$\sqrt{(rp_{x1} - rp_{x2})^2 + (rp_{y1} - rp_{y2})^2} = d \tag{22}$$

gives us an equation to compute r

$$r = \frac{d}{\sqrt{(p_{x1} - p_{x2})^2 + (p_{y1} - p_{y2})^2}} \,. \tag{23}$$

Note that r is positive, therefore, we do not have to deal with signs.

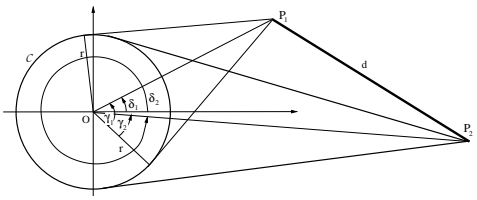


Figure 7: Calibration from a relative distance d between two points $\mathbf{P_1}$ and $\mathbf{P_2}$.

2.2 Extrinsic Parameters of the Right Symmetric 360×360 Mosaic Camera

Like in a case of a central camera, we can define a camera centered Cartesian coordinate system for the right symmetric 360×360 mosaic camera. We place the origin of this coordinate system into the center of rotation and orient its axes so that the x and y axes lie in the plane of rotation δ, i. e. the plane of the circle $\mathcal{C}$, and the z axis points upwards along the axis of rotation. Assume a Cartesian scene coordinate system, which can be related to the camera centered one by a rotation and a translation. This relation is determined by six parameters of a rigid motion and they are the extrinsic parameters of the right symmetric 360×360 mosaic camera.

2.3 Right Symmetric 360×360 Mosaic Camera Model recovery

Provided with known coordinates of scene points and their images, the right symmetric 360×360 mosaic camera parameters can be estimated. From a mosaic image pair, we can compute 3D point coordinates in the camera centered coordinate system. The only intrinsic parameter, the radius r, relates these coordinates to the true ones by an isotropic scaling. Then we look for a Euclidean transformation relating these coordinates to the known scene point coordinates.

Denoting coordinates of points in the right symmetric 360×360 mosaic camera centered coordinate system as $\mathbf{x}_i = (x_i, y_i, z_i, 1)^T$, the relation between them and the coordinates in the scene coordinate system $\tilde{\mathbf{x}}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i, 1)^T$ can be generally expressed as:

$$\mathbf{x}_i = \mathbf{H}\tilde{\mathbf{x}}_i \ . \tag{24}$$

The 4×4 similarity transformation matrix $\mathbf{H}$ represents both intrinsic and extrinsic right symmetric 360×360 mosaic camera parameters and can be decomposed into matrices $\mathbf{K}$ and $\mathbf{M}$, $\mathbf{H} = \mathbf{KM}$, where

$$\mathbf{K} = \begin{pmatrix} r & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{25}$$

contains the right symmetric 360×360 mosaic intrinsic camera parameter, the radius r, and the matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{26}$$

contains the extrinsic parameters, where the 3×3 matrix $\mathbf{R}$ stands for the rotation and the 3×1 vector $\mathbf{t}$ for the translation. We can use for example the method from (Umeyama, 1991) to obtain a least-square estimate of the rotation, translation, and radius.

3 PRACTICAL SITUATION — 360×360 MOSAIC CAMERA

In practice, the physical assembly of the 360×360 mosaic camera may introduce some deviations from the ideal case so that the 1D omnidirectional camera is no longer perpendicular to δ and tangent to $\mathcal{C}$. Then, the image formation model for the right symmetric 360×360 mosaic camera does not hold. In fact, when the tangency assumption does not hold, there is no epipolar geometry. We can only define search curves for each point in both images in the mosaic image pair (Huang et al., 2001), but these curves are not projected mutually onto each other. Anyway, the general 360×360 mosaic camera can be modeled when we take into an account the rotations that break the tangency (by ω) and the perpendicularity (by φ) as it is depicted in Figure 8.

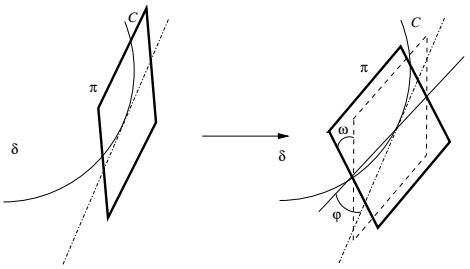


Figure 8: Rotation of the pencil π by the angles φ and ω from its ideal position. The rotation by φ breaks tangency to $\mathcal{C}$, while the rotation by ω breaks perpendicularity to δ, the plane containing $\mathcal{C}$.

3.1 Rotation by ω

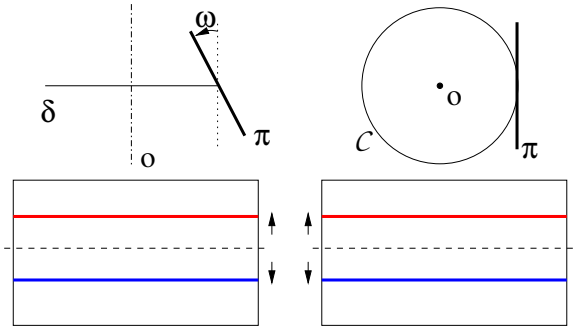


Figure 9: A rotation by ω, when the planar pencil of light rays π is not perpendicular to the plane δ containing the circle of rotation $\mathcal{C}$, but when π is still tangent to $\mathcal{C}$, results in a shift of epipolar lines by the same factor and direction in both mosaic images.

The rotation by ω breaks the perpendicularity of the 1D omnidirectional camera to the plane δ but preserves the tangency to $\mathcal{C}$. Linear epipolar geometry is also preserved, but the rows are transformed nonlinearly compared to rows in images from the right symmetric 360×360 mosaic camera, see Figure 9.

3.2 Rotation by φ

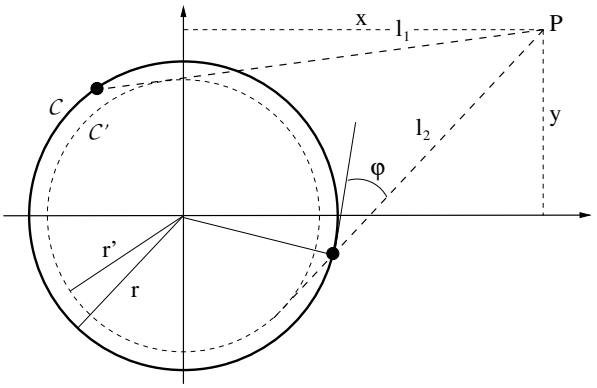


Figure 10: Rotation of the plane π by an angle φ from its original position. results in different lengths of the light rays l_1 and l_2 by which the 360×360 mosaic camera observes a scene point $\mathbf{P}$.

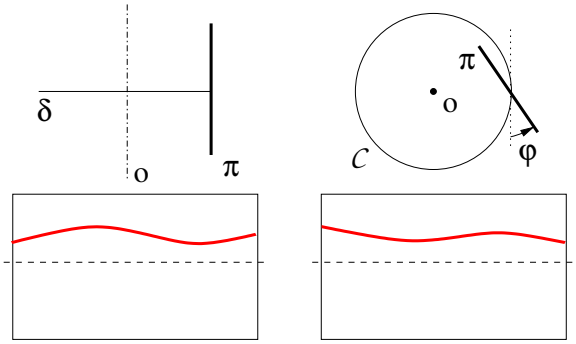


Figure 11: A rotation by φ, when π is not tangent to $\mathcal{C}$ but it is perpendicular to δ, breaks the epipolar geometry. Corresponding points do not lie on corresponding epipolar curves but on search curves instead.

The rotation by φ breaks the tangency to $\mathcal{C}$ while preserving perpendicularity to δ. Moreover, the light rays l_1, l_2, by which a scene point is observed, do not have the same length, as can be seen in Figure 10. Therefore, the angle β is not the same for the two views of the same scene point, like in the case of the (right) symmetric 360×360 mosaic camera, and thus the rotation by φ does not preserve epipolar rows. We have two angles β_φ and β'_φ, one for each view. In fact, there is no epipolar geometry in this case, we can only define some search curves, one for each point (Huang et al., 2001), as it is depicted in Figure 11.

Imagine that we have the same scene points observed by both the right symmetric 360×360 mosaic camera (under angles α'_φ, β_φ, and β'_φ) and the 360×360 mosaic camera (under angles α, α', and β placed at the same position in the scene coordinate system. We can see that using basic trigonometric rules and identities, we can compute the values of angles α_φ, α'_φ, β_φ, and β'_φ based on α, α', and β. We can write:

$$\Delta d = r \sin(\varphi), \tag{27}$$

$$\Delta\alpha = \arcsin\left(\frac{\Delta d}{r}\right) = \varphi, \tag{28}$$

$$\alpha_\varphi = \alpha + \Delta\alpha, \tag{29}$$

$$\alpha'_\varphi = \alpha' + \Delta\alpha, \quad (30)$$

$$\beta_\varphi = \arctan\left(\frac{z}{d-\Delta d}\right), \quad (31)$$

$$\beta'_\varphi = \arctan\left(\frac{z}{d+\Delta d}\right) . \quad (32)$$

We can rewrite the above equations as:

$$\frac{1}{\tan\beta_\varphi} = \frac{d-\Delta d}{z} = \frac{d}{z} - \frac{\Delta d}{z} = \frac{1}{\tan\beta} - \frac{\Delta d}{z} \quad (33)$$

$$\frac{1}{\tan\beta'_\varphi} = \frac{d+\Delta d}{z} = \frac{d}{z} + \frac{\Delta d}{z} = \frac{1}{\tan\beta} + \frac{\Delta d}{z} \quad (34)$$

We can now see that β can be expressed as

$$\beta = \frac{\beta_\varphi + \beta'_\varphi}{2} . \quad (35)$$

Therefore, we can compute the 3D reconstruction of the scene points as

$$x = r\frac{\cos\delta_\varphi}{\cos\gamma_\varphi} \quad (36)$$

$$y = r\frac{\sin\delta_\varphi}{\cos\gamma_\varphi} \quad (37)$$

$$z = r\tan\gamma_\varphi\tan\beta , \quad (38)$$

where r is the radius of the viewing circle, $\gamma_\varphi = \frac{\alpha_\varphi - \alpha'_\varphi}{2}$, and $\delta_\varphi = \frac{\alpha_\varphi + \alpha'_\varphi}{2}$.

The projection from the scene to the mosaic image can be computed first by using equations (9)—(15) and then applying equations (28)—(32) on the image coordinates.

3.3 In-plane rotation by $\Delta\beta$

This is a special case, when the 1D omnidirectional camera is rotated inside of the plane π. This rotation does not affect the geometry of the right symmetric 360×360 mosaic camera, it just changes the image rows, moves a row up in one image and its corresponding row in the second image down, by the same factor, see Figure 12. Only one point correspondence is required for rectification of the mosaic image pair (Bakstein and Pajdla, 2001) by simple shift of the image rows, see Figure 12.

3.4 Combination of all rotations of the 1D omnidirectional camera

In practice, the plane π is rotated by ω, φ, and $\Delta\beta$ at once. We measure some α, α', β, and β' in the mosaic images, but these angles are affected by the rotations of π, see Figure 13.

We can again compensate for the rotation by φ during 3D reconstruction by averaging the row coordinates of a corresponding point in the two mosaic images. This also compensates for the rotation by $\Delta\beta$. The 3D point coordinates

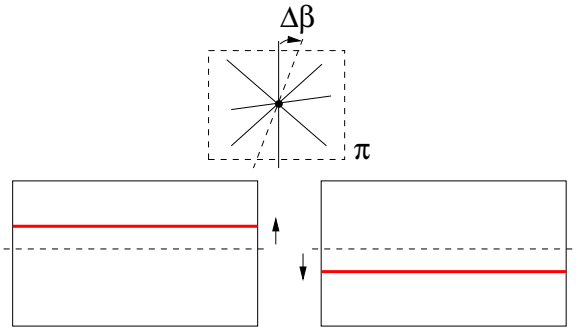


Figure 12: An in-plane rotation results in shifting of image rows in opposite way by the same factor. A linear epipolar geometry is preserved.

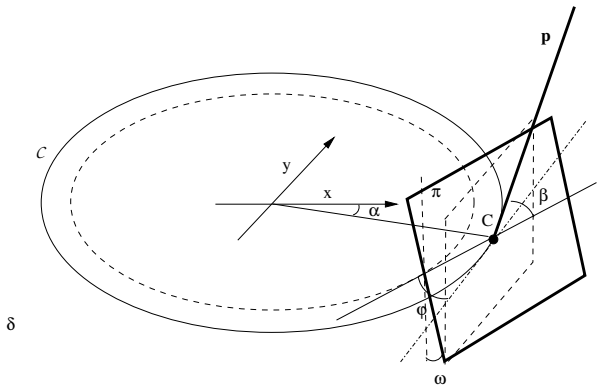


Figure 13: Rotation of the pencil π by both angles ω and φ from its original position. See text for a description.

can then be computed as

$$\Delta\alpha = \tan^{-1}(\tan\beta\sin\omega) , \quad (39)$$

$$r' = r\cos\Delta\alpha , \quad (40)$$

$$\gamma' = \gamma - \Delta\alpha , \quad (41)$$

$$x = r'\frac{\cos\delta}{\cos\gamma'} , \quad (42)$$

$$y = r'\frac{\sin\delta}{\cos\gamma'} , \quad (43)$$

$$l = \sqrt{x^2+y^2} , \quad (44)$$

$$d = \sqrt{l^2+r^2-2r\cos\gamma} , \quad (45)$$

$$z' = d\tan\beta , \quad (46)$$

$$z = z'\cos\omega . \quad (47)$$

The projection from the scene into mosaic images is as follows.

$$l = \sqrt{x^2+y^2} \quad (48)$$

$$\gamma = \arccos\left(\frac{r}{l}\right) \quad (49)$$

$$\delta = \arcsin\left(\frac{y}{l}\right) \quad (50)$$

$$\Delta d = r\sin(\varphi) \quad (51)$$

$$z_p = \frac{z}{\cos(\omega)} \quad (52)$$

$$d_z = z_p\sin(\omega) \quad (53)$$

$$
\begin{aligned}
d_p &= \sqrt{l^2 - (d_z - r)^2} && (54)\\
d &= \sqrt{d_p^2 + d_z^2} && (55)\\
\Delta\alpha &= \arcsin\left(\frac{d_z}{d}\right) && (56)\\
\alpha &= \delta + \gamma - \Delta\alpha - \varphi && (57)\\
\alpha' &= \delta - \alpha + \Delta\alpha - \varphi && (58)\\
\beta &= \arctan\left(\frac{z_p}{d - \Delta d}\right) - \Delta\beta && (59)\\
\beta' &= \arctan\left(\frac{z_p}{d + \Delta d}\right) + \Delta\beta && (60)\\
& && (61)
\end{aligned}
$$

Again, we can describe the intersection of the light ray L and the plane $\boldsymbol{\pi}$ for both rotations as

$$
\begin{aligned}
\mathbf{C} &= (r\cos(\alpha-\varphi), r\sin(\alpha-\varphi), 0) && (62)\\
\mathbf{p} &= (-\cos(\alpha-\varphi)\sin\omega\sin\beta + \\
&\quad \sin(\alpha-\varphi)\cos\beta, \\
&\quad \sin(\alpha-\varphi)\sin\omega\sin\beta + \\
&\quad \cos(\alpha-\varphi)\cos\beta, \\
&\quad \cos(\omega)\sin\beta) && (63)\\
\boldsymbol{\pi} &= (\cos(\alpha'-\varphi), \sin(\alpha'-\varphi), \\
&\quad \sin\omega, r\cos\Delta\alpha\cos\omega)\ , && (64)
\end{aligned}
$$

where $\Delta\alpha = \arctan(\tan\beta\sin\omega)$.

3.5 Bundle adjustment

We cannot recover values of the rotations by ω, φ, and $\Delta\beta$ directly from the mosaic images only, we have to employ a so called bundle adjustment approach, where we optimize for the camera parameters together with 3D coordinates of reconstructed points. As a quality measure we use a distance between measured $\hat{\mathbf{u}}$ image points and images points $\mathbf{u}$ computed by reprojection of the 3D points back into images. The goal of the bundle adjustment is to find scene points $\mathbf{X}$ and the 360×360 mosaic camera parameters ω, φ, and $\Delta\beta$ that minimize this distance. In other words, we solve the following criteria function

$$
J(\omega, \varphi, \Delta\beta, \mathbf{X}) = \min_{\omega,\varphi,\Delta\beta,\mathbf{X}} \sum_n \|\mathbf{u} - \hat{\mathbf{u}}\|\ , \qquad (65)
$$

where n is the number of points.

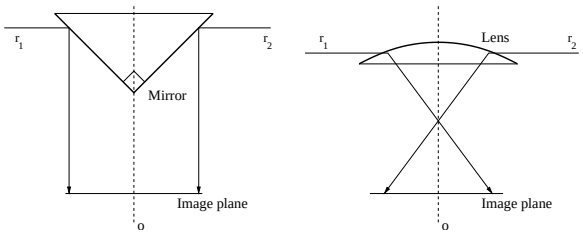
Note that we omitted the radius r because it affects the 3D coordinates by isotropic scaling and therefore its value cannot be recovered without some prior knowledge of the scene. There is also a difference in bundle adjustment for classical pinhole cameras and the 360×360 mosaic camera. In the former case, we estimate a projection matrix using some minimal set of points and the remaining points exhibit an error randomly distributed in both image axes. In the latter case, we can only measure a difference in row coordinates caused by the rotation of the 1D omnidirectional camera by φ, the 3D points will all be projected back onto the same column coordinates. Despite of this, the row difference is enough to start the bundle adjustment, summarized in Algorithm 1.

Algorithm 1 Bundle adjustment

1. Initialize values of the radius r and the angles ω, φ, and $\Delta\beta$.
2. Reconstruct 3D coordinates of points $\mathbf{X}$ using a method of intersection of a plane $\boldsymbol{\pi}$ and a line L, see (62)—(64). Use the initial estimate of the angles ω, φ, and $\Delta\beta$.
3. Optimize (65) over the 3D coordinates of points and all the 360×360 mosaic camera parameters so that the geometric image error is minimized.

4 PRACTICAL REALIZATION OF THE 1D OMNIDIRECTIONAL CAMERA

Originally (Nayar and Karmarkar, 2000), mirrors were proposed for practical realization of the 1D omnidirectional camera, see Figure 14(a), where a telecentric lens observes a conical mirror with 90° at the apex. However, the optical axis of the camera has to be aligned with the axis of rotation of the mirror. Moreover, this system captures only parallel light rays and does not allow any compensation for rotations by ω and φ. We have proposed use of a fisheye lens with field of view larger than 180° (Bakstein and Pajdla, 2001), illustrated in Figure 14(b). Fisheye lens reduces the problem of alignment and moreover, it captures light rays with various angles, not only those in a plane tangent to $\mathcal{C}$. In case of rotations by ω and φ, we can use these light rays to create right symmetric 360×360 mosaic camera from imperfectly composed setup.



(a) (b)

Figure 14: (a) Light rays reflected by the conical mirror are orthographically projected onto a circle. (b) Light rays in one plane are projected by the fish eye lens onto a circle on the sensor.

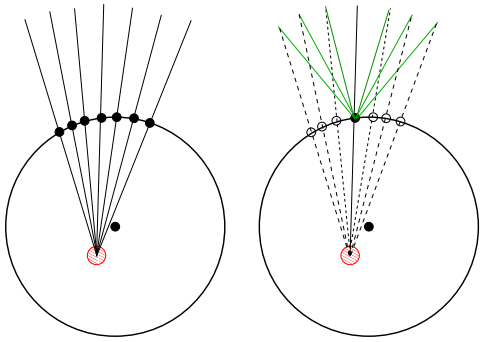
5 APPLICATION — IMAGE BASED RENDERING

Image based rendering (IBR) is a representation of a real environment by a sequence of images. A virtual camera can be moved in some constrained regions and light rays from the original sequence of images are used to approximate image from this virtual camera. There are several approaches, summarized in (Shum and Kang, 2000), which differ in requirement of some assumption on geometry of the scene and in dimensionality of the subset of the plenoptic function captured. We prefer X-slits rendering (Zomet et al., 2003) because it captures a 3D subset of the plenoptic function offering a reasonable trade-off between a freedom of movement in a virtual representation of the real

environment and amount of data required for its representation.

But even a 3D subset of the plenoptic function means huge memory requirement. It has been shown (Bakstein and Pajdla, 2004), that the number of input images, and thus the memory requirements, can be greatly reduced using only mild assumptions on the scene depth. The right symmetric 360×360 mosaic camera can be used to acquire such a depth estimate thanks to its linear epipolar geometry. Moreover, the practical realization of the right symmetric 360×360 mosaic camera captures input images for the X-slits rendering at the same time as it acquires mosaic images of the scene, when a fisheye lens is used instead of a mirror.

The reduction of the memory requirements is done by decimating the input sequence for the IBR. Instead of using all required images, we take only a subset of images. Then, we have to approximate rays from the omitted images by rays from the nearest image in the sequence, see Figure 15. This ray approximation requires a scene depth assumption. Fortunately, the depth can be estimated with a tolerance depending on the scene distance and the number of images in the decimated sequence. Figure 16 shows depth ranges of good approximation for a given number of images and estimated scene depth. It can be observed that closer part of the scene needs much more precise depth estimate than distant part of the scene.



(a) (b)

Figure 15: The same part of the scene can be covered by (a) large number of images or (b) respective number of light rays captured by one image.

6 EXPERIMENTAL RESULTS

At first we demonstrate, that a fisheye lens can be accurately modeled. Figure 17 shows the fitting and the prediction error for our model of the Nikon FC-E8 fisheye lens, which can capture up to 183° field of view. The model can be described by the following function (Bakstein and Pajdla, 2002)

$$r = a \tan \frac{\theta}{b} + c \sin \frac{\theta}{d}, \qquad (66)$$

where θ is the angle between a light ray and the optical axis and r is an image distance from the optical center. It can be seen that the prediction error is very low. When

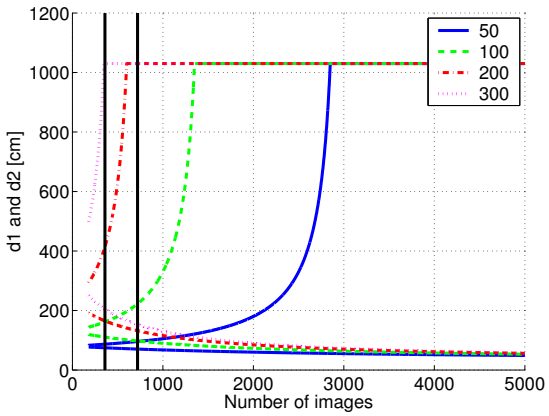


Figure 16: Depth ranges in which the scene will be rendered correctly for a given number of images and a normalization depth.

measured in pixels, the error will depend on size and resolution of a CCD chip used. We can conclude that we are able to model the 1D omnidirectional camera by selecting appropriate pixels from a fisheye image.

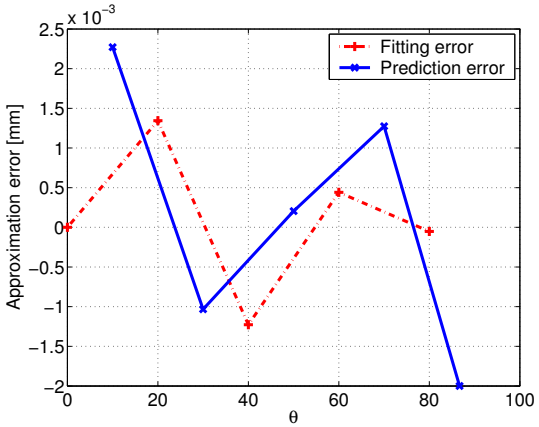


Figure 17: Fitting and prediction error for the proposed model for the FC-E8 lens.

To demonstrate the bundle adjustment procedure, we selected manually 16 points in the mosaic images. Even though we tried to compose the 360×360 camera setup as precisely as we could, corresponding points in the two mosaic images were not on the same image rows, that means they were not observed under the same β angles. It can be observed in Figure 18 that the reprojection error in the mosaic images was reduced by the bundle adjustment under one pixel.

A depth map recovered from a 360×360 mosaic image, see Figure 1, is depicted in Figure 19. It can be noticed that all the relevant scene depth information was recovered, including the closest object (the desk with the keyboard). This allows faithful rendering of the scene with a reasonably small number of input images for the IBR. Figure 20 shows output image from the IBR without (a) and with (b) the depth estimate. A clear improvement in the latter case can be observed.

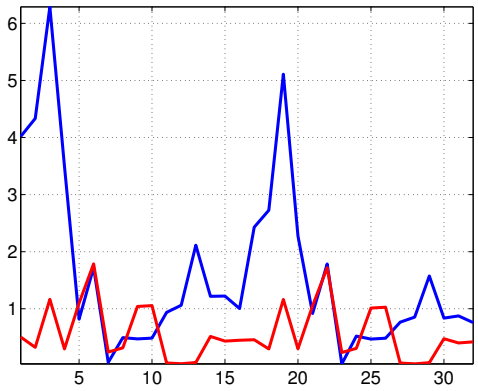


Figure 18: Bundle adjustment. The blue curve denotes the geometric error for the initial estimate of the 360×360 mosaic camera parameters. The red curve shows the error after the bundle adjustment.

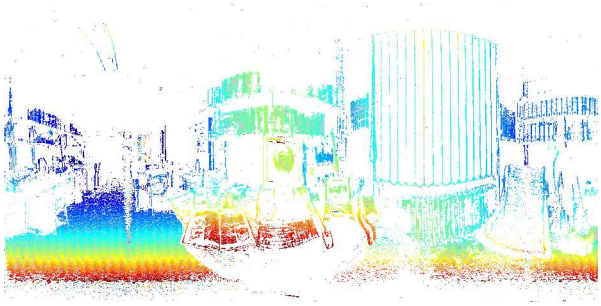
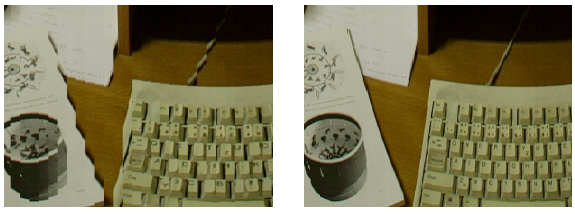


Figure 19: Depth map recovered from a right symmetric 360×360 mosaic image.

7 CONCLUSION

We have identified different geometries of the mosaic camera, namely the right symmetric 360×360 mosaic camera, with linear epipolar geometry and a single intrinsic parameter, and a general 360×360 mosaic camera which takes into account imperfections in the physical realization employing a 1D omnidirectional camera rotated on a circular path. We presented geometric models for both geometries together with a method for direct recovery of parameters of the right symmetric 360×360 mosaic camera and a bundle adjustment procedure for the 360×360 mosaic camera. Moreover, we discussed use of fisheye lenses for a practical realization of the 1D omnidirectional camera. Finally, we have presented an application of the right symmetric 360×360 mosaic camera in image based rendering.



(a) (b)

Figure 20: IBR output without scene depth knowledge (a) and with estimated scene depth (b). Note missing features in the former case.

8 ACKNOWLEDGMENTS

The authors were supported by The Czech Science Foundation under project GACR 102/03/0440, by The FP5 EU under project IST-2001-39184, by The Czech Ministry of Education under projects MSMT Kontakt 22-2003-04 and Kontakt ME 678, by The STINT under project Dur IG2003-2 062 and by The Austrian Ministry of Education under project CONEX GZ 45.535.

REFERENCES

Bakstein, H. and Pajdla, T., 2001. 3D reconstruction from 360 x 360 mosaics. In: A. Jacobs and T. Baldwin (eds), Proceedings of the CVPR'01 conference, Vol. 2, pp. 72–77.

Bakstein, H. and Pajdla, T., 2002. Panoramic mosaicing with a $\mathbf{180}^\circ$ field of view lens. In: A. D. Williams (ed.), Proceedings of the IEEE Workshop on Omnidirectional Vision, pp. 60–67.

Bakstein, H. and Pajdla, T., 2004. Visual fidelity of image based rendering. In: D. Skočaj (ed.), Proceedings of the Computer Vision Winter Workshop 2004 (CVWW'04), Slovenia, pp. 139–148.

Baratoff, G. and Aloimonos, Y., 1998. Changes in surface convexity and topology caused by distortions of stereoscopic visual space. In: ECCV98, pp. 226–240.

Huang, F., Wei, S. K. and Klette, R., 2001. Geometrical fundamentals of polycentric panoramas. In: J. Little and D. Lowe (eds), ICCV'01: Proc. Eighth IEEE International Conference on Computer Vision, Vol. 1, USA, pp. 560–565.

Huang, F., Wei, S.-K. and Klette, R., 2003. Comparative studies of line-based panoramic camera calibration. In: Proceedings of Omnivis 2003: Workshop on Omnidirectional Vision and Camera Networks, IEEE Press, p. cdrom only.

Nayar, S. K. and Karmarkar, A., 2000. 360 x 360 mosaics. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00), Hilton Head, South Carolina, Vol. 2, pp. 388–395.

Shum, H.-Y. and Kang, S. B., 2000. A review of image-based rendering techniques. In: IEEE/SPIE Visual Communications and Image Processing (VCIP) 2000, pp. 2–13.

Shum, H.-Y., Kalai, A. and Seitz, S. M., 1999. Omnivergent stereo. In: Proc. of the International Conference on Computer Vision (ICCV'99), Kerkyra, Greece, Vol. 1, pp. 22–29.

Umeyama, S., 1991. Least-squares estimation of transformation parameters between two point patterns. PAMI 13(4), pp. 376–380.

Zomet, A., Feldman, D., Peleg, S. and Weinshall, D., 2003. Mosaicing new views: The crossed-slits projection. IEEE Trans. on Pattern Analysis and Machine Intelligence 25(6), pp. 741–754.