

# A Factorization Based Self-Calibration for Radially Symmetric Cameras

Srikumar Ramalingam, Peter Sturm, Edmond Boyer

► **To cite this version:**

Srikumar Ramalingam, Peter Sturm, Edmond Boyer. A Factorization Based Self-Calibration for Radially Symmetric Cameras. International Symposium on 3D Data Processing, Visualization and Transmission, 2006, Chapel Hill, United States. 2006. <inria-00384315>

**HAL Id: inria-00384315**

**<https://hal.inria.fr/inria-00384315>**

Submitted on 14 May 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Factorization Based Self-Calibration for Radially Symmetric Cameras

Srikumar Ramalingam<sup>1&2</sup>, Peter Sturm<sup>1</sup> and Edmond Boyer<sup>1</sup>

<sup>1</sup> INRIA Rhône-Alpes, GRAVIR-CNRS, 38330 Montbonnot, France

<sup>2</sup> Dept. of Computer Science, University of California, Santa Cruz, CA 95064, USA

{Srikumar.Ramalingam, Peter.Sturm, Edmond.Boyer}@inrialpes.fr

## Abstract

The paper proposes a novel approach for planar self-calibration of radially symmetric cameras. We model these camera images using notions of distortion center and concentric distortion circles around it. The rays corresponding to pixels lying on a single distortion circle form a right circular cone. Each of these cones is associated with two unknowns; optical center and focal length (opening angle). In the central case, we consider all distortion circles to have the same optical center, whereas in the non-central case they have different optical centers lying on the same optical axis. Based on this model we provide a factorization based self-calibration algorithm for planar scenes from dense image matches. Our formulation provides a rich set of constraints to validate the correctness of the distortion center. We also propose possible extensions of this algorithm in terms of non-planar scenes, non-unit aspect ratio and multi-view constraints. Experimental results are shown.

## 1 Introduction and Previous Work

Recently many unconventional camera models have been studied by vision researchers for different applications. The most popular ones are the omnidirectional cameras with large fields of views. These cameras can be constructed using only lenses (dioptric) or using a combination of mirrors and lenses (catadioptric) [2, 25, 5]. Tailor made calibration algorithms exist for all these models. In contrast there has been some recent work on adopting a very generic imaging model and calibrating it. This model considers every camera as an unconstrained set of projection rays [20, 7, 14, 15]. Here, calibration consists in determining the non-parametric mapping between pixels and associated projection rays. These algorithms may perform well in practice, however they require the knowledge of the scene structure. Self-calibration has been recently shown to be possible for constrained camera motions [17, 16], but mainly as proof of concept. It will remain quite involved to make it stable in practice.

In reality a completely generic camera does not exist. Most cameras are symmetric in some way. The most com-

mon form is radial symmetry. Practically used cameras such as pinhole and fisheye are all radially symmetric (modulo a non-unit aspect ratio). Most catadioptric configurations are radially symmetric. For example, when the optical center of the pinhole camera is located on the axis of revolution of the mirror, the resulting configuration is radially symmetric.

In this work we propose a self-calibration algorithm for radially symmetric cameras using two or more views of unknown planar scenes. There are few works along this direction [23, 24, 9]. The most closely related work is by Tardif and Sturm [23], which uses the same camera model, but the algorithm uses known calibration grids. Thirthala and Pollefeys [24] propose a linear solution for recovering radial distortion which can also include non-central cameras. Every radial line passing through the distortion center is mapped to coplanar rays. As a consequence their algorithm involves projective reconstruction followed by metric upgradation, using dual of the absolute conic, for self-calibration. Hartley and Kang [9] proposed a planar calibration algorithm for correcting radial distortion. They provide a method to compute the distortion center. However, their model is again restricted to central cameras and the usage of known calibration grids

Our approach maps every distortion circle around the distortion center to a cone of rays. We transform the self-calibration problem to a neat factorization framework requiring only a singular value decomposition using dense image matches. Dense matching is still an overhead, but in practice, it is possible to interpolate sparse matches on planar scenes. We also provide a rich set of constraints to validate or to even estimate the distortion center. There are several other works for estimating radial distortion, but these use more specific distortion models and/or rely on the extraction of line images [26, 6, 25, 10, 28, 29].

**Organization:** We formally introduce our camera model and the problem statement in section 2. Next the factorization framework for the self-calibration problem will be provided for non-central and central models. Then the possible variants such as the usage of non-planar scenes, non-unit aspect ratio and usage of more than three views are discussed. In the final section we discuss the experiments.

## 2 Problem Definition

### 2.1 Camera Model

We model radially symmetric cameras using a unique *distortion center* and concentric distortion circles centered about this point as shown on the left side of Figure 1. The projection rays of pixels lying on a specific distortion circle form a right viewing cone (see the right side of Figure 1). These viewing cones may have different focal lengths (opening angles) and optical centers. However all the optical centers lie on the same line – the optical axis – which also corresponds to the axis of all viewing cones. We assume the  $z$  axis to be the optical axis. The viewing cones intersect the  $xy$  plane in concentric circles. Variables  $d_i$  are used to parameterize the positions of optical centers and, together with radii  $r_i$ , the focal lengths. Note that this model is sufficiently general to include both central and non-central radially symmetric cameras.

For simplicity we translate the coordinate system of the image such that the distortion center becomes the origin. Let us consider a pixel on the distortion circle with radius  $\check{r}$  at an angle  $\theta$ . Its location can be specified by  $(\check{r}\cos(\theta), \check{r}\sin(\theta))$ . The corresponding 3D ray in the viewing cone is specified by the optical center  $(0, 0, d)$  and the point  $(r\cos(\theta), r\sin(\theta), 0)$  in the  $xy$  plane.

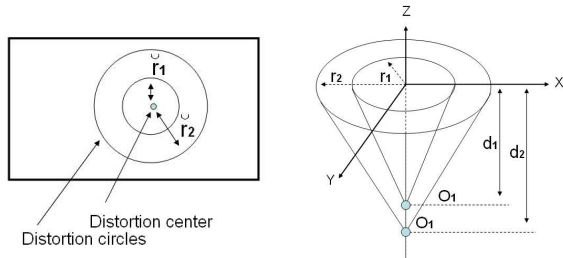


Figure 1: Radially symmetric camera model. Left: Distortion center and two distortion circles. Right: Corresponding viewing cones. They may have different vertices (optical centers) and opening angles.

### 2.2 Problem Statement

This work focuses on the planar-based self-calibration of radially symmetric cameras. Two or three views of a planar scene are captured from different camera locations. The input is the dense image matches between these views. The goal is to compute the distortion center, optical centers ( $d_i$ ) and opening angles ( $r_i/d_i$ ) of the cones associated with each of the distortion circles. In other words we are interested in computing the projection rays associated with every pixel in a radially symmetric image.

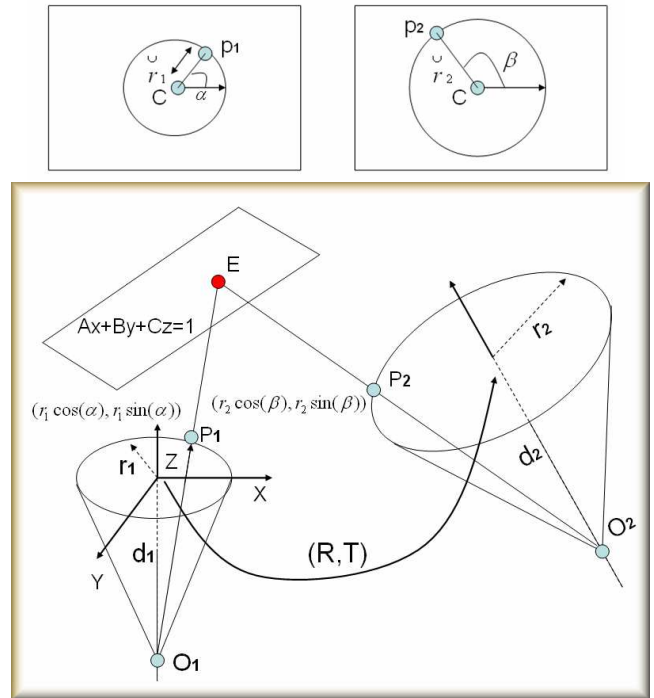


Figure 2: Top: Two matching pixels  $p_1 = (r_1 \cos(\alpha), r_1 \sin(\alpha))$  and  $p_2 = (r_2 \cos(\beta), r_2 \sin(\beta))$  in the images. Bottom: Triangulation of the corresponding rays  $O_1P_1$  and  $O_2P_2$ , coming from two different cameras, intersecting at a point  $E$  on the plane.

## 3 Algorithm: Factorization Framework

In the following we give the derivation of the self-calibration algorithm. In the beginning we assume that the distortion center is known. Later in section 3.3, we provide a technique to compute the distortion center. Consider the corresponding images shown on the top of Figure 2. Let  $p_1 = (r_1 \cos(\alpha), r_1 \sin(\alpha))$  and  $p_2 = (r_2 \cos(\beta), r_2 \sin(\beta))$  be two matching image pixels.

The main idea behind the self-calibration comes from the popular triangulation constraint on the intersection of projection rays corresponding to the matching pixels. In 3D space we define the first camera coordinate system as shown on the right side of Figure 1. Usually optical center will be chosen for the origin. Since we have several optical centers, corresponding to different viewing cones, the optical axis of the camera is chosen as the  $z$  axis. The individual optical centers of the cones lie at a distance of  $d_i$  from the origin. We chose  $xy$  plane to intersect all the viewing cones and we parameterize the individual  $r_i$  on it. Now consider the scenario at the bottom of the Figure 2, where we intersect the corresponding 3D rays. The projection rays start from

the respective optical centers and project towards the 3D points  $(r_1 \cos(\alpha), r_1 \sin(\alpha), 0)$  and  $(r_2 \cos(\beta), r_2 \sin(\beta), 0)$  respectively.

Let the equation of the planar scene be given by:  $Ax + By + Cz = 1$  in the coordinate system of the *second* camera. Let  $(R, T)$  be the motion between the first and the second camera positions. The parameters of the plane  $(A, B, C)$ , motion  $(R, T)$ , two focal lengths  $(d_1/r_1, d_2/r_2)$  and optical centers  $(d_1, d_2)$  are all unknown. The only information we have is the fact that the matching projection rays intersect at a point on the plane. We use this constraint to solve the unknowns and eventually calibrate the camera. The second projection ray starts from  $(0, 0, d_2)$  and passes through the point  $(r_2 \cos(\beta), r_2 \sin(\beta), 0)$ . Now we compute the point of intersection of this ray with the plane, represented by the equation  $Ax + By + Cz = 1$ . We obtain the intersection point  $\check{E}$  in the second coordinate system:

$$\check{E} = \begin{pmatrix} r_2 \cos(\beta)(1 - Cd_2) \\ r_2 \cos(\beta)(1 - Cd_2) \\ d_2(Ar_2 \cos(\beta) + Br_2 \sin(\beta) - 1) \\ Ar_2 \cos(\beta) + Br_2 \sin(\beta) - Cd_2 \end{pmatrix}$$

On expressing  $\check{E}$  in the first coordinate frame we obtain  $E$  (cf. Figure 2).

$$E = \begin{pmatrix} R & T \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} r_2 \cos(\beta)(1 - Cd_2) \\ r_2 \cos(\beta)(1 - Cd_2) \\ d_2(Ar_2 \cos(\beta) + Br_2 \sin(\beta) - 1) \\ Ar_2 \cos(\beta) + Br_2 \sin(\beta) - Cd_2 \end{pmatrix}$$

We introduce a set of intermediate variables, coupling camera motion and plane coefficients:

$$\begin{aligned} a_{1,1} &= R_{1,1} + AT_1 \\ a_{1,2} &= R_{1,2} + BT_1 \\ a_{2,1} &= R_{2,1} + AT_2 \\ a_{2,2} &= R_{2,2} + BT_2 \\ a_{3,1} &= R_{3,1} + CT_3 \\ a_{3,2} &= R_{3,2} + CT_3 \\ b_{1,1} &= -CR_{1,1} + AR_{1,3} \\ b_{1,2} &= -CR_{1,2} + BR_{1,3} \\ b_{1,3} &= -R_{1,3} - CT_1 \\ b_{2,1} &= -CR_{2,1} + AR_{2,3} \\ b_{2,2} &= -CR_{2,2} + BR_{2,3} \\ b_{2,3} &= -R_{2,3} - CT_2 \\ b_{3,1} &= -CR_{3,1} + AR_{3,3} \\ b_{3,2} &= -CR_{3,2} + BR_{3,3} \\ b_{3,3} &= -R_{3,3} - CT_3 \end{aligned}$$

Using the above notations we denote  $E$  as follows:

$$\begin{pmatrix} r_2 \cos(\beta)(a_{1,1} + d_2 b_{1,1}) + r_2 \sin(\beta)(a_{1,2} + d_2 b_{1,2}) + b_{1,3} d_2 \\ r_2 \cos(\beta)(a_{2,1} + d_2 b_{2,1}) + r_2 \sin(\beta)(a_{2,2} + d_2 b_{2,2}) + b_{2,3} d_2 \\ r_2 \cos(\beta)(a_{3,1} + d_2 b_{3,1}) + r_2 \sin(\beta)(a_{3,2} + d_2 b_{3,2}) + b_{3,3} d_2 \\ Ar_2 \cos(\beta) + Br_2 \sin(\beta) - Cd_2 \end{pmatrix} \quad (1)$$

This point must lie on the projection ray associated with pixel  $p_1$  in the first view, i.e. it must be collinear with the optical center  $O_1 = (0, 0, d_1)$ , and the point  $P_1 = (r_1 \cos(\alpha), r_1 \sin(\alpha), 0)$  (cf. figure 2). Collinearity of three points means that when stacking their homogeneous coordinates in a  $4 \times 3$  matrix, the determinants of all four submatrices of size  $3 \times 3$  must be zero. In our case, one of them is always zero and the other three give conditions that are algebraically dependent. One of them are thus used in this work, cf. Equation 2. Note that the following equation is independent of  $r_1$  and  $d_1$ .

$$\begin{aligned} &\cos(\alpha) \cos(\beta) \left( \left( \frac{r_2}{d_2} \right) a_{2,1} + r_2 b_{2,1} \right) + \\ &\cos(\alpha) \sin(\beta) \left( \left( \frac{r_2}{d_2} \right) a_{2,2} + r_2 b_{2,2} \right) - \\ &\sin(\alpha) \cos(\beta) \left( \left( \frac{r_2}{d_2} \right) a_{1,1} + r_2 b_{1,1} \right) - \\ &\sin(\alpha) \sin(\beta) \left( \left( \frac{r_2}{d_2} \right) a_{1,2} + r_2 b_{1,2} \right) + \\ &\cos(\alpha) b_{2,3} - \sin(\alpha) b_{1,3} = 0 \end{aligned} \quad (2)$$

Let us consider a specific distortion circle with radius  $\check{r}_2$  in the second image. We select five pixels in this distortion circle and denote the corresponding matches as follows:  $(\check{r}_2 \cos(\beta_i), \check{r}_2 \sin(\beta_i)), (\check{r}_1^i \cos(\alpha_i), \check{r}_1^i \sin(\alpha_i)), i = 1..5$ . The matching pixels from the second image have the same  $\check{r}_2$ . This implies that the corresponding viewing cone in 3D space has fixed  $r_2$  and  $d_2$ . On the other-hand the radii of the pixels from the first image  $\check{r}_1^i$  are all different and consequently their corresponding viewing cones are also different. However equation 2 does not depend on the viewing cones of the distortion circles from the first image. In other words the equation is independent of  $r_1^i$  and  $d_1^i$ . As a result the only unknowns in our system are  $r_2, d_2, a$  and  $b$ . Note that  $\alpha_i$  and  $\beta_i$  are already known, though they differ for every match. For brevity let us denote  $\cos(\alpha)$  by  $c\alpha$  and  $\sin(\alpha)$  by  $s\alpha$ . Using this notation we obtain the following linear system from equation 2.

$$\Gamma = \begin{pmatrix} c\alpha_1 c\beta_1 & -c\alpha_1 s\beta_1 & -s\alpha_1 c\beta_1 & s\alpha_1 s\beta_1 & c\alpha_1 & -s\alpha_1 \\ c\alpha_2 c\beta_2 & -c\alpha_2 s\beta_2 & -s\alpha_2 c\beta_2 & s\alpha_2 s\beta_2 & c\alpha_2 & -s\alpha_2 \\ c\alpha_3 c\beta_3 & -c\alpha_3 s\beta_3 & -s\alpha_3 c\beta_3 & s\alpha_3 s\beta_3 & c\alpha_3 & -s\alpha_3 \\ c\alpha_4 c\beta_4 & -c\alpha_4 s\beta_4 & -s\alpha_4 c\beta_4 & s\alpha_4 s\beta_4 & c\alpha_4 & -s\alpha_4 \\ c\alpha_5 c\beta_5 & -c\alpha_5 s\beta_5 & -s\alpha_5 c\beta_5 & s\alpha_5 s\beta_5 & c\alpha_5 & -s\alpha_5 \end{pmatrix} \quad (3)$$

$$\Sigma = \begin{pmatrix} ((\frac{r_2}{d})a_{2,1} + r_2b_{2,1}) \\ ((\frac{r_2}{d})a_{2,2} + r_2b_{2,2}) \\ ((\frac{r_2}{d})a_{1,1} + r_2b_{1,1}) \\ ((\frac{r_2}{d})a_{1,2} + r_2b_{1,2}) \\ b_{2,3} \\ b_{1,3} \end{pmatrix} \quad (4)$$

$$\Gamma_{5 \times 6} \times \Sigma_{6 \times 1} = 0 \quad (5)$$

As described above, the matrix  $\Gamma_{5 \times 6}$  is completely known because it involves only the already known  $\alpha_i$  and  $\beta_i$ . On solving the above homogeneous linear system we compute  $\Sigma$  and thereby the values of  $((\frac{r_2}{d})a_{2,1} + r_2b_{2,1})$ ,  $((\frac{r_2}{d})a_{2,2} + r_2b_{2,2})$ ,  $((\frac{r_2}{d})a_{1,1} + r_2b_{1,1})$ ,  $((\frac{r_2}{d})a_{1,2} + r_2b_{1,2})$ ,  $b_{2,3}$  and  $b_{1,3}$  up to a scale. The sixth variable  $b_{1,3}$ , as described earlier as  $(-R_{13} - CT_1)$ , depends only on pose  $(R, T)$  and the plane parameter  $C$ , i.e.  $b_{1,3}$  is independent of the calibration parameters. Thus we fix this value to a constant for all the radii to obtain their corresponding equations in the same scale. We rewrite this information in the following form where  $k_i$ 's are known.

$$\left(\frac{r}{d}\right) r \begin{pmatrix} a_{2,1} & a_{2,2} & a_{1,1} & a_{1,2} \\ b_{2,1} & b_{2,2} & b_{1,1} & b_{1,2} \end{pmatrix} = (k_1 \quad k_2 \quad k_3 \quad k_4)$$

We iterate the above process for different distortion circles, having different radii  $r_2^i$ , and obtain the following.

$$\begin{pmatrix} \left(\frac{r_2^1}{d_1}\right) r_2^1 \\ \left(\frac{r_2^2}{d_2}\right) r_2^2 \\ \vdots \\ \left(\frac{r_2^n}{d_n}\right) r_2^n \end{pmatrix} \begin{pmatrix} a_{2,1} & a_{2,2} & a_{1,1} & a_{1,2} \\ b_{2,1} & b_{2,2} & b_{1,1} & b_{1,2} \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ \vdots & \vdots & \vdots & \vdots \\ k_{n1} & k_{n2} & k_{n3} & k_{n4} \end{pmatrix} \quad (6)$$

$$L_{n \times 2} M_{2 \times 4} = K_{n \times 4} \quad (7)$$

The matrix  $K$  is known up to a scale and both  $L$  and  $M$  are unknowns. Since  $L$  and  $M$  are both of rank 2. We use singular value decomposition to compute the factors of  $K$ .

$$K_{n \times 4} = U_{n \times 4} S_{4 \times 4} V_{4 \times 4}^T$$

Since  $K$  is of rank 2,  $S$  must have only two nonzero singular values. Thus  $S$  must be of the following form.

$$S = \begin{pmatrix} l_1 & 0 & 0 & 0 \\ 0 & l_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}_{4 \times 4}$$

We remove the last few rows we obtain  $\check{U}_{n \times 2}$ ,  $\check{S}_{2 \times 2}$  and  $\check{V}_{s \times 2}$  from the matrices  $U$ ,  $S$  and  $V$  respectively.

$$K = \check{U}_{n \times 2} \check{S}_{2 \times 2} \check{V}_{2 \times 2}^T$$

The matrices  $L$  and  $M$  can be computed up to 4 unknowns ( $X_{2 \times 2}$ ) as given below:

$$L = U X_{2 \times 2}, \quad M = X_{2 \times 2}^{-1} V$$

The framework is very similar to the one used in [27]. Without loss of generality we can fix the distance of the optical center of the first distortion circle to be 1 unit from the origin. This implies that we can reduce one degree of freedom in  $X$ . We use an approximate technique to compute the remaining three variables. This is done using two assumptions. First we assume that the ratio of the radii of very small distortion circles, near the center of the image, is equal to the ratio of the radii of their corresponding viewing cones. Second assumption is that the optical centers of the viewing cones of very small distortion circles coincide. It is important to understand the significance of these four parameters. These four parameters represent the scale factor in the computation of the focal length and optical centers. In addition, there is a translational ambiguity in the estimation of the optical centers along the optical axis. The three unknowns can also be computed analytically using the additional images. However for practical radially symmetric non-central cameras (such as spherical catadioptric cameras) this approximation works fine. The interesting part of this framework is that the internal and external parameters are disambiguated in the calibration process. Once we compute  $X$  we can compute  $L$  and  $M$  uniquely. The computation of  $L$  provide us the necessary calibration information. As given in equation 7, we obtain  $r_i$  and  $d_i$  on the computation of  $L$ . This will provide us the cone of rays for every distortion circle and eventually the calibration in the form of mapping between image pixels and projection rays.

### 3.1 Central cameras

In the case of the central cameras, all distortion circles are associated with the same optical center. This implies that there will be a single  $d$  for all viewing cones. We can fix this to be unity. The equation 2 simplifies itself to the following form.

$$\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix} (a_{2,1} + b_{2,1} \quad a_{2,2} + b_{2,2} \quad a_{1,1} + b_{1,1} \quad a_{1,2} + b_{1,2}) = \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ \vdots & \vdots & \vdots & \vdots \\ k_{n1} & k_{n2} & k_{n3} & k_{n4} \end{pmatrix}$$

We do a singular value decomposition of  $K$  to obtain  $U$ ,  $S$  and  $V$ . The rank of  $K$  is unity. This implies that the individual radii can be computed up to a common scale. By fixing the radius of one distortion circle to unity we can compute the other radii. The overall scale can be computed using

more images. We briefly explain this procedure. We capture two images of a general 3D scene. We compute a  $3 \times 3$  fundamental matrix with respect to the projection rays. This is equivalent to computing the fundamental matrix between the distortion corrected images. From the fundamental matrix we can extract the focal length [21].

### 3.2 Geometrical Interpretation

The constraint given in equation 2 is extremely rich. For example we can consider a central model where  $\alpha = \frac{\pi}{2}$ . The corresponding equation is given below.

$$(a_{1,1} + b_{1,1})r_2c\beta + (a_{1,2} + b_{1,2})r_2s\beta + b_{1,3} = 0$$

This considers all possible matches for different values of  $r_1$ ,  $r_2$  and  $\beta$ . In the first image this constraint corresponds to considering all pixels lying on the negative  $y$  axis (see Figure 3). On the second image this refers to the equation of a line. This implies that in a radially symmetric central camera the mapping of a radial line passing through the distortion center is a line. Note that in the Figure 3 the mapping in the second image is a distorted line. This is because the distortion center was not fixed correctly. Thus by imposing this constraint we can check for the correctness of the distortion center. In the next subsection we will see how to use such constraints to compute the distortion center. In the case of non-central case we have the following.

$$c\beta(r_2a_{1,1} + r_2d_2b_{1,1}) + s\beta(r_2a_{1,2} + r_2d_2b_{1,2}) + b_{1,3}d_2 = 0$$

The above equation again represents a line in the first image. However it is not mapped to a line in the second image. Similarly we can obtain several constraints corresponding to other scenarios ( $\alpha = 0, \beta = 0, \alpha\beta = 0, \alpha\beta = 1, r_1 = r_2$ , etc). We show some of these constraints which are obtained from equation 2 in Table 1 for simple cases in a central model.

Cases	Equation 2
$r_2 = 0$	$b_{2,3}\cos\alpha + b_{1,3}\sin\alpha = 0$
$\alpha = 0$	$r_2\cos\beta(a_{2,1} + b_{2,1}) + r_2\sin\beta(a_{2,2} + b_{2,2}) + b_{2,3} = 0$
$\alpha = \frac{\pi}{2}$	$(a_{1,1} + b_{1,1})r_2\cos\beta + (a_{1,2} + b_{1,2})r_2 + b_{1,3} = 0$
$\beta = 0$	$\cos\alpha(r_2(a_{2,1} + b_{2,1}) + b_{2,3}) + \sin\alpha(r_2(a_{1,1} + b_{1,1}) + b_{1,3}) = 0$
$\beta = \frac{\pi}{2}$	$\cos\alpha(r_2(a_{2,2} + b_{2,2}) + b_{2,3}) + \sin\alpha(r_2(a_{1,2} + b_{1,2}) + b_{1,3}) = 0$

Table 1: Some constraints obtained from Equation 2 for specific values of  $\alpha$ ,  $\beta$  and  $r_2$

### 3.3 Computation of the distortion center

In this section we will briefly explain our algorithm to verify the correctness of the distortion center. We fix the distortion

center at the center of the image for fisheye and pinhole images. In the case of catadioptric cameras with full image of the mirror boundary, observed as a conic, we can initialize the distortion center at the center of the conic. From the general equation 2 we learned how to compute  $r$  and  $d$ . We validate the correctness of the matches by substituting in the general equation 2. In addition by expanding another  $3 \times 3$  sub-matrix of the Matrix 1 we obtain the following equation.

$$\begin{aligned} &\cos(\alpha)\cos(\beta)\left(\left(\frac{r_1}{d_1}\right)\left(\frac{r_2}{d_2}\right)a_{3,1} + \left(\frac{r_1}{d_1}\right)r_2b_{3,1} + r_1\left(\frac{r_2}{d_2}\right)c_{1,1}\right) + \\ &\cos(\alpha)\sin(\beta)\left(\left(\frac{r_1}{d_1}\right)\left(\frac{r_2}{d_2}\right)a_{3,2} + \left(\frac{r_1}{d_1}\right)r_2b_{3,2} + r_1\left(\frac{r_2}{d_2}\right)c_{1,2}\right) + \\ &\cos(\alpha)\left(\left(\frac{r_1}{d_1}\right)b_{3,3} + r_1c_{1,3}\right) + \\ &\cos(\beta)\left(\left(\frac{r_2}{d_2}\right)a_{1,1} + r_2b_{1,1}\right) + \\ &\sin(\beta)\left(\left(\frac{r_2}{d_2}\right)a_{1,2} + r_2b_{1,2}\right) + b_{1,3} = 0 \end{aligned}$$

In addition to equation 2 we use the above equation in checking the correctness of the distortion center. We compute the error we obtain on using the different solutions of  $r_1, d_1, r_2$  and  $d_2$ . We compute the overall error using all the matches. The correct distortion center will be the point which minimizes this error. This technique can also be used to compute the distortion center starting from the center of the image and computing the error at various points surrounding it.

## 4 Variants

### 4.1 Non-planar scenes

Using the earlier parameterization the intersection point can be computed on the first and the second rays in the first camera coordinate system:

$$P_1 = \begin{pmatrix} \lambda_1 r_1 \cos\alpha \\ \lambda_1 r_1 \sin\alpha \\ d_1 - \lambda_1 d_1 \\ 1 \end{pmatrix} \quad P_2 = \begin{pmatrix} R & T \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \lambda_2 r_2 \cos\beta \\ \lambda_2 r_2 \sin\beta \\ d_2 - \lambda_2 d_2 \\ 1 \end{pmatrix}$$

$\lambda_1$  and  $\lambda_2$  are two parameters used in locating the intersection point in the two rays. By matching  $P_1$  and  $P_2$  we obtain three equations. Two equations are sufficient to eliminate  $\lambda_1$  and  $\lambda_2$ . The third equation could be used to solve for the calibration parameters.

### 4.2 Non-unit aspect ratio

In the presence of non-unit aspect ratio,  $\gamma$ , we can model the projection ray of every pixel  $p(\check{r}\cos\theta, \check{r}\sin\theta)$  to the projection ray passing through  $(0, 0, d_1)$  and  $(r\cos\theta, \gamma r\sin\theta, 0)$ .

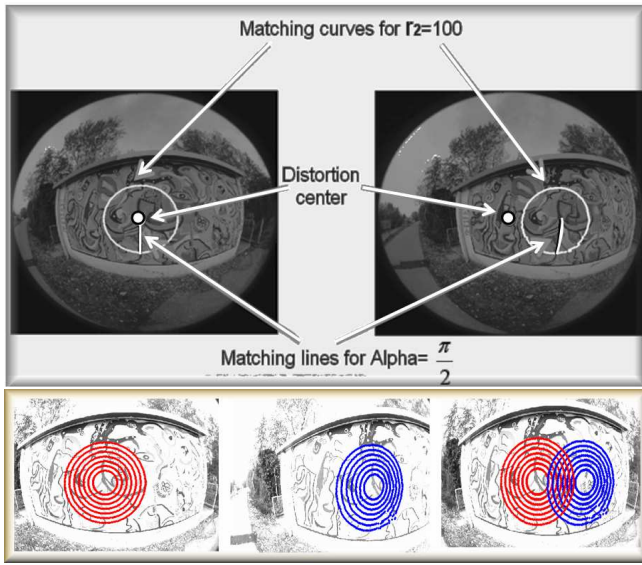


Figure 3: Top: Two planar scenes captured by fisheye lenses. In polar coordinates the matches are represented by  $(\check{r}_1, \alpha)$  and  $(\check{r}_2, \beta)$ . We show two matching curves under the constraints of  $r_2 = 100$  and  $\alpha = \frac{\pi}{2}$  respectively. Bottom: left: The pixels corresponding to specific values of  $r$ . middle: The matching pixels in the second image. right: We show the matching pixels from both the images. Note that the curves need not intersect. Also note the matching pixels do not form straight lines (refer to their relationship in Equation 2).

Similar to the earlier scenario we can again construct a factorization framework to calibrate the camera.

$$\begin{pmatrix} \left(\frac{r_1}{d_1}\right) & r_1 \\ \left(\frac{r_2}{d_2}\right) & r_2 \\ \cdot & \cdot \\ \left(\frac{r_n}{d_n}\right) & r_n \end{pmatrix} \begin{pmatrix} a_{2,1} & \gamma a_{2,2} & \gamma a_{1,1} & \gamma^2 a_{1,2} \\ b_{2,1} & \gamma b_{2,2} & \gamma b_{1,1} & \gamma^2 b_{1,2} \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ \cdot & \cdot & \cdot & \cdot \\ k_{n1} & k_{n2} & k_{n3} & k_{n4} \end{pmatrix} \quad (8)$$

In the case of known aspect ratio the problem is extremely simple. In the case of unknown aspect ratio we can compute it along with the pose. We look at the non-unit aspect ratio in a different way. Every pixel  $p(\check{r}_1 \cos(\theta), \check{r}_1 \sin(\theta))$  maps to a projection ray passing through  $(0, 0, d_1)$  and  $(r_1 \cos(\theta), \gamma r_1 \sin(\theta))$ . An easier way to understand this would be to think of every circle in the image to map to a cone which is crushed along one axis (distorted cone with an ellipsoidal base). By doing this we avoid the aspect ratio problem in matching features in the image planes.

### 4.3 Multi-View Relations

The factorization framework is easily extendable to multiple views. The constraints from multiple views can be used in the same framework as follows for central cameras. The extension for non-central cameras is also straight-forward.

$$\begin{pmatrix} r_1 \\ r_2 \\ \cdot \\ r_n \end{pmatrix} (a_{2,1} + b_{2,1} \quad a_{2,2} + b_{2,2} \quad a_{1,1} + b_{1,1} \dots \\ a_{1,2} + b_{1,2} \quad a'_{2,1} + b'_{2,1} \dots a'_{2,2} + b'_{2,2}) = \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k'_{11} & k'_{12} & k'_{13} & k'_{14} & \dots \\ k_{21} & k_{22} & k_{23} & k_{24} & k'_{21} & k'_{22} & k'_{23} & k'_{24} & \dots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \dots \\ k_{n1} & k_{n2} & k_{n3} & k_{n4} & k'_{n1} & k'_{n2} & k'_{n3} & k'_{n4} & \dots \end{pmatrix}$$

where  $a_{i,j}, b_{i,j}, k_{ij}$  are associated with first and the second views. We extend this with  $a'_{i,j}, b'_{i,j}, k'_{ij}$ , which can be associated with second and third or any other two views.

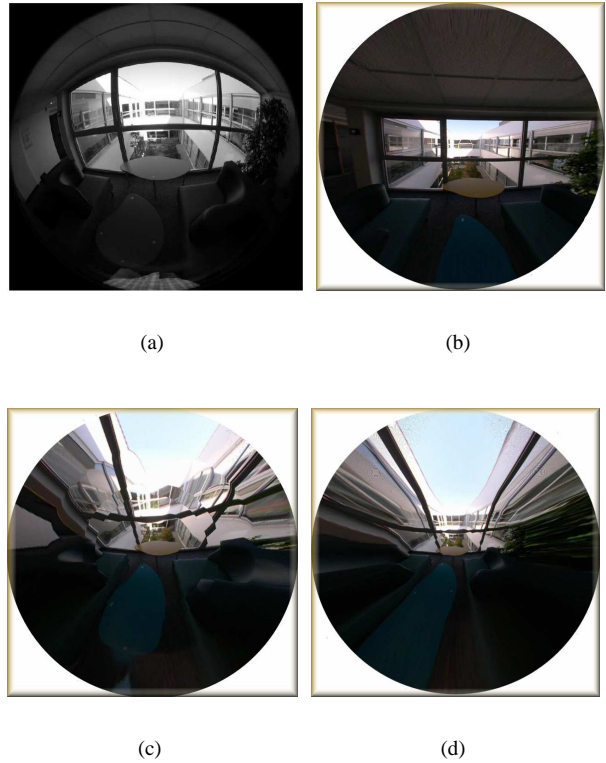


Figure 4: Distortion correction (perspective view synthesis). (a) Original fisheye images (b) Using the distortion center at the correct location. (c) The distortion center is at an offset of 25 units from the correct distortion center. (d) The distortion center is at a distance of 50 units from the correct position. The image size is 1024 by 768.



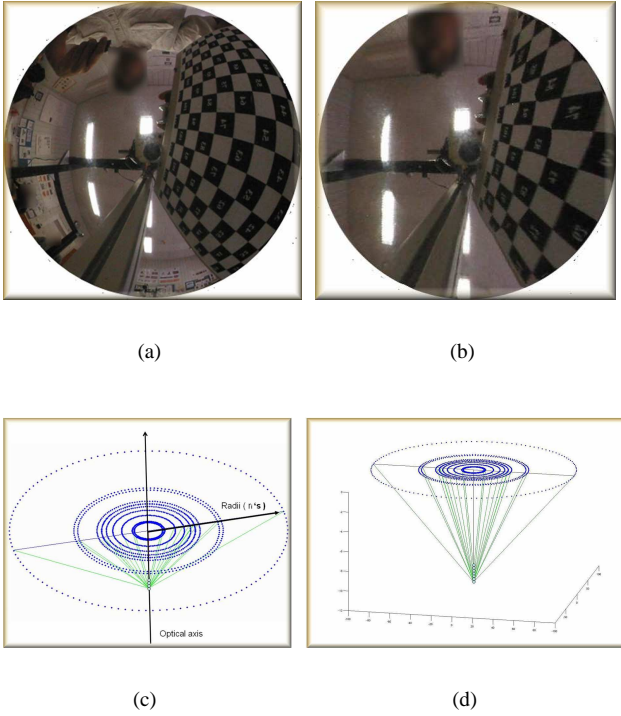


Figure 5: (a) Image taken by a spherical catadioptric camera. (b) Distortion correction. Note that the camera model is non-central and distortion correction is not possible. We compute an approximate center close to all the projection rays and performed distortion correction. (c) and (d) show the reconstructed projection rays for spherical catadioptric camera. Note that we do not show the distortion correction for the complete image. This is because a single image pair of the plane was not sufficient to calibrate the complete image. By using more than two images we can calibrate the whole image. This is similar in spirit to [15] and the multi-view constraints are given in section 4.3.

## 5 Experiments

### 5.1 Cameras

We tested our algorithms on three different cameras: Nikon coolpix 5400, E8 fisheye lens with a field of angle of 182 by 360 degrees, a non-central catadioptric camera with a spherical mirror. We modeled all three cameras as radially symmetric.

### 5.2 Dense matching in planar scenes

There are plenty of planar scenes in man made environments such as walls, posters, etc. In addition the matching process is simplified in planar scenarios. For perspective

cameras points undergo a homography transformation between two views of a planar scene. This allows us to obtain dense matching from initial set of sparse matches. For the case of fisheye lenses we used Harris corner detection to detect features on a plane. Then image matching is done using cross-correlation based approach. This worked well for planes which are not very close to the fisheye lens. In Figure 3 we detected 270 feature matches. Using them we can interpolate for other matches. We found that the cubic interpolation to be very precise for planar scenes observed by fisheye cameras. We also used planar boards with black dots to simplify the matching process. These objects were used for catadioptric cameras with spherical mirrors.

### 5.3 Distortion correction

We used distortion correction to study the accuracy of our algorithm. In Figure 4 we show an undistorted original fisheye image and the distortion corrected images. Once we estimate the parameters of the viewing cone the process of distortion correction is very simple. We briefly explain the distortion correction procedure. Every pixel with coordinates  $(\check{r}\cos\theta, \check{r}\sin\theta)$  will be moved to  $(r\cos\theta, r\sin\theta)$  where  $r$  is the radius of the associated viewing cone. Note that this correction can not be applied for non-central images. In the central cases the rank of matrix  $K$  (refer Equation 7) must be 1. However in general the matrix was found to have a higher rank. So RANSAC must be used to improve the stability of the algorithm. It is applied in selecting five correspondences in every distortion circle in the second image.

**Sensitivity of the distortion center:** We estimate the distortion center as explained in section 3.3. In Figure 4(c) and (d) we show the distortion correction when the distortion center was placed at a distance of 25 and 50 pixels respectively. The problem that appear in the two images are very common if the distortion center is not fixed at the correct location. In Figure 4(a) the relative radii are not computed correctly. This can be seen as artifacts at concentric circles. In the second case (Figure 4(b)) the outer radii explode. This kind of errors happen in wide angle scenarios. Both these errors don't take place on using RANSAC and the choice of correct distortion center.

### 5.4 Non-central model

We used a catadioptric camera with a spherical mirror to study the non-central model. As per the theory we obtained a  $K$  matrix of rank 2 (refer Equation 7). We did a singular value decomposition to get the first two columns of the  $U$  matrix. However as we studied earlier the solution is obtained in terms of three variables. Analytical computation of these three variables involves usage of the additional conditions arising from the collinearity constraint and more



views of the plane. Here we used an approximate method to compute the three variables. First we assume that the radial distortion is negligible near the center of the image. This means that the ratio of the radii of two distortion circles is the same as the ratio of radii of its corresponding viewing cones. Please note that this assumption is very close to the correct ratio for fisheye cameras and central catadioptric cameras. The second assumption is that the optical centers of two viewing cones will be the same near the center of the image. Using these two assumptions we were able to compute the three unknown variables. Once the unknown variables were computed, we reconstruct the projection rays. In Figure 5 (c) and (d) we show the reconstructed projection rays. Note that the camera is non-central and all the projection rays intersect at the optical axis. On using the exact algorithm to compute the re-projection rays we expect the rays to correlate well with the actual caustics of this catadioptric configuration. The optical centers of all the projection rays are already quite close, which is in fact the case of spherical catadioptric configuration. We also tested distortion correction of an image taken by spherical catadioptric camera (see Figure 5). Since the model is not a central one we first compute an approximate center for all the projection rays. Considering the fact that we are using a non-central camera for distortion correction the algorithm performs reasonably well. The distortions in the calibration grid, corners of the wall, etc are removed (see Figure 5(b)). These are only preliminary results in our approach to validate our theory. We intend to use non-linear optimization to improve the accuracy.

## 6 Conclusions

We propose a simple method to solve the planar self-calibration problem for radially symmetric cameras, which can be both, central or non-central. The theoretical formulation is sufficiently general for extending to non-planar scenes and cameras with non-unit and unknown aspect ratio. Most of our efforts went in obtaining a neat factorization framework for the self-calibration problem. Though the initial experimental results are promising, we consider this work to be a theoretical contribution rather than a practical one. In future we plan to conduct more experiments on the extensions of this framework to non-planar scenes and multiple views.

**Acknowledgments:** This work was supported by a grant from the European Community under the EST Marie-Curie project Visitor.

## References

- [1] D. Aliaga. Accurate Catadioptric Calibration for Real-size Pose Estimation of Room-size Environments, *ICCV*, 2001.
- [2] S. Baker, S. Nayar. A theory of catadioptric image formation. *ICCV*, 1998.
- [3] H. Bakstein, T. Pajdla. An overview of non-central cameras. *Computer Vision Winter Workshop*, Ljubljana, Slovenia, 2001.
- [4] D. Feldman, T. Pajdla, D. Weinshall. On the Epipolar Geometry of the Crossed-Slits Projection. *ICCV*, 2003.
- [5] C. Geyer, K. Daniilidis. A unifying theory of central panoramic systems and practical implications. *ECCV*, 2000.
- [6] A.W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. *CVPR*, 2001.
- [7] M.D. Grossberg, S.K. Nayar. A general imaging model and a method for finding its parameters. *ICCV*, 2001.
- [8] R. Gupta, R.I. Hartley. Linear Pushbroom Cameras. *PAMI* 1997.
- [9] R. Hartley, S.B. Kang. A parameter free method for estimating radial distortion. *ICCV*, 2005.
- [10] B. Micusik, T. Pajdla. Autocalibration and 3D Reconstruction with Non-central Catadioptric Cameras. *CVPR*, 2004.
- [11] J. Neumann, C. Fermüller, Y. Aloimonos. Polydioptric Camera Design and 3D Motion Estimation. *CVPR*, 2003.
- [12] T. Pajdla. Stereo with oblique cameras. *IJCV*, 2002.
- [13] S. Peleg, M. Ben-Ezra, Y. Pritch. Omnistere: Panoramic Stereo Imaging. *PAMI*, 2001.
- [14] R. Pless. Using Many Cameras as One. In *CVPR*, 2003.
- [15] S. Ramalingam, P. Sturm, S.K. Lodha. Towards Complete Generic Camera Calibration. *CVPR*, 2005.
- [16] S. Ramalingam, P. Sturm, S.K. Lodha. Towards Generic Self-Calibration of Central Cameras. *OMNIVIS*, 2005.
- [17] D. Nister, H. Stewenius, E. Grossmann, Non-parametric Self-Calibration. *ICCV*, 2005.
- [18] S. Ramalingam, P. Sturm, S.K. Lodha. Theory and calibration algorithms for axial cameras. *ACCV*, 2006.
- [19] S. Seitz, J. Kim. The Space of All Stereo Images. *IJCV*, 2002.
- [20] P. Sturm, S. Ramalingam. A generic concept for camera calibration. *ECCV*, 2004.
- [21] P. Sturm, Z. Cheng, P.C.Y. Chen, A.N. Poo. Focal length calibration from two views: method and analysis of singular cases. *CVIU*, 2005.
- [22] R. Swaminathan, M.D. Grossberg, S.K. Nayar. A perspective on distortions. *CVPR*, 2003.
- [23] J.P. Tardif, P. Sturm. Calibration of Cameras with Radially Symmetric Distortion. *OMNIVIS*, 2005.
- [24] S. Thirthala, M. Pollefeys. 1D radial cameras and its application to omnidirectional geometry. *ICCV*, 2005.
- [25] D. Claus, A.W. Fitzgibbon. A rational function model for fish-eye lens correction. *CVPR*, 2005.
- [26] R. Swaminathan and S. Nayar. Non-metric calibration of wide-angle lenses and polycameras. *CVPR*, 1999.
- [27] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 1992.
- [28] D.C. Brown. Close-Range Camera Calibration. *Photometric Engineering*, 37(8), 855-866, 1971.
- [29] F. Devernay and O. Faugeras. Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments. *MVA*, 2001.