Self-Calibration of a General Radially Symmetric Distortion Model

Jean-Philippe Tardif, Peter Sturm, Sébastien Roy

HAL Id: inria-00387130

https://hal.inria.fr/inria-00387130

Submitted on 24 May 2009

Self-calibration of a general radially symmetric distortion model

Jean-Philippe Tardif[1], Peter Sturm[2], and Sébastien Roy[1]

[1] DIRO, Université de Montréal,Canada.
{`tardifj, roys`}`@iro.umontreal.ca`
[2] INRIA Rhône-Alpes. 38330 Montbonnot St Martin, France.
{`Peter.Sturm`}`@inrialpes.fr`

Abstract. We present a new approach for self-calibrating the distortion function and the distortion center of cameras with general radially symmetric distortion. In contrast to most current models, we propose a model encompassing fisheye lenses as well as catadioptric cameras with a view angle larger than 180°.

Rather than representing distortion as an image displacement, we model it as a varying focal length, which is a function of the distance to the distortion center. This function can be discretized, acting as a general model, or represented with e.g. a polynomial expression.

We present two flexible approaches for calibrating the distortion function. The first one is a plumbline-type method; images of line patterns are used to formulate linear constraints on the distortion function parameters. This linear system can be solved up to an unknown scale factor (a global focal length), which is sufficient for image rectification. The second approach is based on the first one and performs self-calibration from images of a textured planar object of unknown structure. We also show that by restricting the camera motion, self-calibration is possible from images of a completely unknown, non-planar scene.

The analysis of rectified images, obtained using the computed distortion functions, shows very good results compared to other approaches and models, even those relying on non-linear optimization.

1. Introduction

Most theoretical advances in geometric computer vision make use of the pin-hole camera model. One benefit of such a model is the linearity of the projection which simplifies multi-view constraints and other structure-from-motion computations. Unfortunately in many cases, this model is a poor representation of how the camera samples the world, especially when dealing with wide angle cameras where radial distortion usually occurs. In addition to these cameras, catadioptric devices (i.e. cameras pointed at a mirror) also admit a very large field of view. Their image distortion can also be seen as a type of radial distortion, although, in general, it cannot be modeled with traditional models. This is because the view angle of these cameras can be larger than 180°, which is not compatible

2

with the usual *image-displacement* approach. The effect of radial distortion is that straight lines in the scene are not in general projected onto straight lines in the image, contrary to pin-hole cameras. Many calibration algorithms can deal with distortion, but they are usually tailor-made for specific distortion models and involve non-linear optimization.

In this paper, we introduce a general distortion model, whose main feature is to consider radially symmetric distortion. More precisely, we make the following assumptions on the camera projection function:

- the aspect ratio is 1,
- the distortion center is aligned with the principal point[3],
- the projection function is radially symmetric (around the distortion center),
- the projection is central, i.e. projection rays pass through a single (effective) optical center.

Given the quality of camera hardware manufacturing, it is common practice to assume an aspect ratio of 1. As for the second and third assumptions, they are made to ensure our model is consistent with both catadioptric devices and regular fisheye cameras. Finally, a central projection is assumed for simplicity even for very large field of view cameras [1, 22] in which a non-single viewpoint might be induced by the lens [3], or by a misaligned mirror [18].

Our full camera model consists therefore of the position of the distortion center and the actual distortion function that maps distance from the distortion center to focal length. This model, together with the above assumptions, fully represents a camera projection function. It is a good compromise between traditional low-parametric camera models and fully general ones, modeling one projection ray per pixel [10, 17], in terms of modeling power and ease and stability of calibration. The model is indeed general enough to represent cameras of different types and with very different view angles.

Problem statement. In this paper, we intend to solve the proposed model relying on images of collinear points in space. Our algorithm makes no assumption on the distortion function and on the distortion center position. Only a rough initial value of the latter is needed.

Organization. A short review of the most popular distortion models is presented in the first section. The model we adopt is presented in §3. In §4 we propose a plumbline method for calibrating our model using images of collinear points. Based on this, we propose a plane-based self-calibration approach, in §5. Finally, the performance of our methods is analyzed and compared to another similar approach [6].

2. Related Work

As the field of view of a camera lens increases, the distortion occurring in the captured images becomes more and more important. Traditionally, researchers

---

[3] We will see that this constraint may be dropped in some cases.

have sought new models with more degrees of freedom and complexity. These models include the traditional polynomial model [11] (which can be combined with a field of view model (FOV) [6]), division [7] and rational [5]. Most of the time the models are calibrated using non-linear optimization of either a full projection model from points located on a calibration object [23] or a homography mapping from a planar grid [5]. Recent papers have also shown that radial distortion models can be calibrated linearly from a calibration grid [12] of by feature point matching between images [7, 5, 19, 20].

Other approaches focus only on calibrating the distortion function by imposing either that a straight line in space should appear straight in the image [4, 6] or that spherical objects should appear circular [16].

The aforementioned models all apply to cameras with a field of view smaller than 180° since the distortion is *image-based*. They fail to handle data captured by a camera with a view angle larger than 180°, typical for catadioptric devices. Different models and algorithm have been specifically designed to address these cases [9, 14] and their parameters have an explicit geometric interpretation rather than expressing distortion directly.

Finally, only few attempts were made to find models able to deal with dioptric systems (including radial distortion) and catadioptric ones [22, 2]. The model we propose fits in this category with the benefit that its distortion function can be general.

3. Camera Model

We describe the camera model that corresponds to the assumptions explained in the introduction. Consider a camera with canonical orientation, i.e. the optical axis is aligned with the Z-axis and image x and y-axes are parallel to world X and Y-axes respectively. Our camera model is then fully described by the position of a distortion center $(c_x, c_y)^\top$ and a distortion "function" $f : \mathcal{R} \to \mathcal{R}$, such that an image point $(x, y)^\top$ is back-projected to a 3D line spanned by the optical center and the point at infinity with coordinates:

$$\left[x - c_x,\, y - c_y,\, f(r),\, 0\right]^\top ,\ \ r = \sqrt{(x - c_x)^2 + (y - c_y)^2}$$

The distortion function (it should actually be called "undistortion function", but we did not find this very elegant) can for example be chosen as a polynomial with even powers of r, in which case we have the division model, as used in [7, 19]. The model also subsumes fisheye models [8, 15] and cameras of the 'unified central catadioptric model' [9].

In this paper, we use two representations for the distortion function. The first one is a polynomial of a degree d to be fixed, like in the division model, however including odd powers:

$$f(r) = \sum_{i=0}^{d} \lambda_i r^i. \tag{1}$$

The second one is a discrete representation, consisting of a lookup table of the distortion function values at a set of discrete values for r (in practice, we use one sample per step of one pixel). We denote these values as:

$$f(r) = f_r. \tag{2}$$

Note that a constant function f allows the representation of a pinhole camera with f's value as focal length. From the above back-projection equation, it is easy to deduce equations for distortion correction, also called rectification in the sequel. This can for example be done by re-projecting the points at infinity of projection rays into a pinhole camera with the same optical center and orientation as the original camera. As for the intrinsic parameters of the (virtual) pinhole camera, we usually also adopt an aspect ratio of 1 and zero skew; if the distortion center is to have the same coordinates in the rectified image as in the original one, and if g denotes the rectified image's focal length, then the homogeneous coordinates of the rectified point are:

$$\begin{bmatrix} g & 0 & c_x \\ 0 & g & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - c_x \\ y - c_y \\ f(r) \end{bmatrix}.$$

In the following, we introduce a few geometric notions that will be used in this paper. A **distortion circle** is a circle in the image, centered in the distortion center. Projection rays of points lying on a distortion circle span an associated **viewing cone** in space. In our model, all cones have the same axis (the optical axis) and vertex (the optical center).

Each cone can actually be understood as an individual pinhole camera, with $f(r)$ as focal length (r being the distortion circle's radius). Geometrically, this is equivalent to virtually moving the image plane along the optical axis, according to the distortion function. This situation is depicted in fig. 1. In the case of a camera with a view angle larger than 180°, the focal length becomes equal or smaller than zero. In the zero case, the cone is actually the **principal plane**, i.e. the plane containing the optical center and that is perpendicular to the optical axis. Let us call the associated distortion circle **principal distortion circle**. A negative $f(r)$ is equivalent to a camera with positive focal length, looking backward and whose image is mirrored in x and y. Typical situations for rectification are depicted in fig. 2.

Rectification for cameras with a view angle larger than 180° cannot be done as usual: the above rectification operation is no longer a bijection (two points in the original image may be mapped to the same location in the rectified one) and points on the principal distortion circle are mapped to points at infinity (fig. 2b). It is still possible to rectify individual parts of the image correctly, by giving the virtual pinhole camera a limited field of view and allowing it to rotate relative to the true camera.

4. Plumbline Calibration

In this section, we show that the distortion function f and the distortion center can be recovered linearly from the images of lines (straight edges) or points that
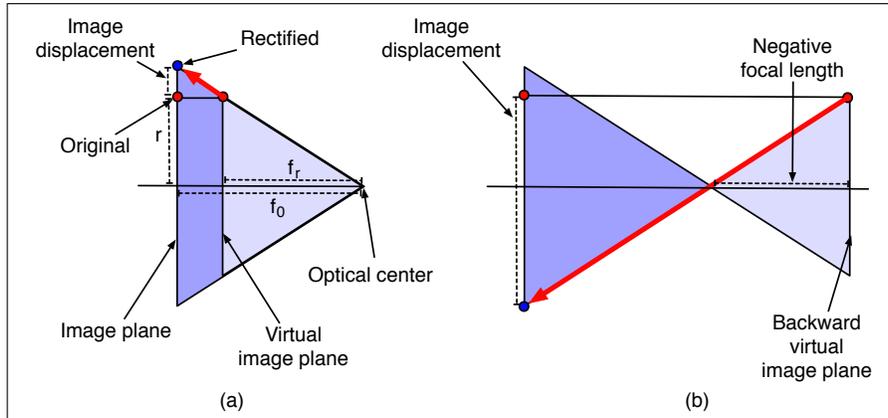
Fig. 1. Distortion circles are associated with cones in space. Theoretically, any point of the image can be projected into a single plane. **a)** Pixel from a cone looking forward, **b)** one from a cone looking backward.

are collinear in space. This is thus akin to the classical plumbline calibration technique [4, 6].

4.1. Calibration of Distortion Function

We obtain linear constraints on the distortion function as follows. Consider the images of three collinear points, $\mathbf{p}_i = (x_i, y_i)^\top$. For now, let us assume that the distortion center is known and that the image coordinate system is centered in this point. Hence, $r_i = ||(x_i, y_i)||$ is the distance of a point from the distortion center. Provided that these points should be collinear once rectified, we know that:

$$\begin{vmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ f(r_0) & f(r_1) & f(r_2) \end{vmatrix} = 0 \tag{3}$$

which can be written explicitly as a linear constraint on the $f(r_i)$'s:

$$f(r_0) \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + f(r_1) \begin{vmatrix} x_2 & x_0 \\ y_2 & y_0 \end{vmatrix} + f(r_2) \begin{vmatrix} x_0 & x_1 \\ y_0 & y_1 \end{vmatrix} = 0. \tag{4}$$

If f is of the form (1) or (2), then this equation gives a linear constraint on its parameters λ_i respectively f_r.

Constraints can be accumulated from all possible triplets of points that are projections of collinear points in space. We thus obtain a linear equation system of the form $\mathbf{A}\mathbf{x} = \mathbf{0}$, where $\mathbf{x}$ contains the parameters of f (the λ_i's or the f_r's). Note that constraints from triplets where two or all three image points lie close to one another are not very useful and hence can be neglected in order to reduce the number of equations. Solving this system to least squares yields parameters that maximize the collinearity of the rectified points[4]. Note that the equation

---

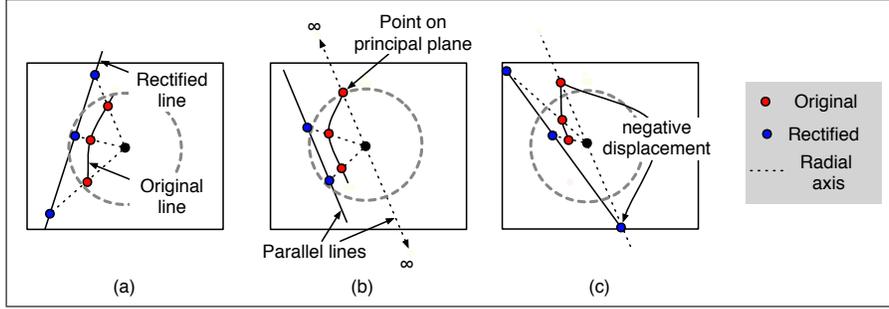[4] However, it is not optimal in terms of geometric distance.

Fig. 2. Situations where three points are rectified into collinear positions. **a)** Three points corresponding to forward cones. **b)** One point located on principal distortion circle, i.e. scene point on principal plane. **c)** Two points on forward cones and one on a backward cone.

system is homogeneous, i.e. the distortion parameters are only estimated up to scale. This is natural, as explained below; a unique solution can be guaranteed by setting $\lambda_0 = 1$ as is usually done for the division model, or by setting one f_r to a fixed value.

4.2. Calibration of Distortion Center

So far, we have assumed a known distortion center. In this section, we show how it can be estimated as well, in addition to the actual distortion function. A first idea is to sample likely positions of the distortion center, e.g. consider a regular grid of points in a circular region in the image center, and compute the distortion function for each of them using the above method. We then keep the point yielding the smallest residual of the linear equation system as the estimated distortion center. This approach is simple and not very elegant, but is fully justified and works well in practice. Its downside is that the computation time is proportional to the number of sampled points.

Therefore, we investigate a local optimization procedure, as opposed to the above brute force one. Let (c_x, c_y) be the unknown distortion center. Equation (3) now becomes:

$$\begin{vmatrix} x_0 - c_x & x_1 - c_x & x_2 - c_x \\ y_0 - c_y & y_1 - c_y & y_2 - c_y \\ f\left(\left\| \begin{bmatrix} x_0 - c_x \\ y_0 - c_y \end{bmatrix} \right\|\right) & f\left(\left\| \begin{bmatrix} x_1 - c_x \\ y_1 - c_y \end{bmatrix} \right\|\right) & f\left(\left\| \begin{bmatrix} x_2 - c_x \\ y_2 - c_y \end{bmatrix} \right\|\right) \end{vmatrix} = 0. \tag{5}$$

First, this constraint cannot be used directly for the discretized version of the distortion function. Second, if we use the polynomial model, the constraint is highly non-linear in the coordinates of the distortion center.

We thus consider an approximation of (5): we assume that a current estimate of the distortion center is not too far away from the true position ($\|(c_x, c_y)\|$ is

small), so that f can be approximated with $(c_x, c_y) = \mathbf{0}$ and

$$f(\left\| \begin{bmatrix} x \\ y \end{bmatrix} \right\|) \approx f(\left\| \begin{bmatrix} x - c_x \\ y - c_y \end{bmatrix} \right\|).$$

Equation (5) thus simplifies to:

$$\begin{vmatrix} x_0 - c_x & x_1 - c_x & x_2 - c_x \\ y_0 - c_y & y_1 - c_y & y_2 - c_y \\ f(\left\| \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right\|) & f(\left\| \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \right\|) & f(\left\| \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \right\|) \end{vmatrix} = 0 \tag{6}$$

which is linear in c_x and c_y. Once again, combining many constraints leads to an over-determined linear equation system. The recovered distortion center may not be optimal because the points are expressed relative to the approximate center and because of the simplification of (5). Hoping that the previous assumptions are applicable, this new center should nevertheless improve our rectification. This estimation is used in a local optimization scheme of alternation type:

0. Initialize the distortion center with e.g. the center of the image.
1. Fix the distortion center and compute the distortion function (§4.1).
2. Fix the distortion function and update the distortion center (§4.2).
3. Go to step 1, unless convergence is observed.

Instead of using the least-squares cost function based on the algebraic distance (3), we also consider a more geometric cost function to judge convergence in step 3. Consider a set of image points belonging to a line image. From the current values of distortion center and function, we compute their projection rays and fit a plane as follows: determine the plane that contains the optical center and that minimizes the sum of (squared) angles with projection rays. The residual squared angles, summed over all line images, give the alternative cost function.

4.3. Discussion

The estimation of distortion center and function is based on an algebraic distance expressing collinearity of rectified image points. Better would be of course to use a geometric distance in the original images; this is possible but rather involved and is left for future work.

We briefly describe what the calibration of the distortion function amounts to, in terms of full metric calibration. First, recall that the distortion function can be computed up to scale only from our input (see §4.1). This is natural: if we have a distortion function that satisfies all collinearity constraints, then multiplying it by a scale factor results in a distortion function that satisfies them as well. This ambiguity means that once the distortion function is computed (up to scale) and the image rectified, the camera can be considered as equivalent to a pinhole camera with unknown focal length, with the difference that the field of view is potentially larger than 180°. Any existing focal length calibration or self-calibration algorithm designed for pinhole cameras can be applied to obtain

a full metric calibration. A direct application of such algorithms can probably use only features that lie inside the principal distortion circle, but it should be possible to adapt them so as to use even fields of view larger than 180°. At this step, the second assumption of §1 can also be relaxed if desired: a full pinhole model, i.e. not only focal length, can in principle be estimated from rectified images.

5. Self-Calibration

We now develop a plane-based self-calibration approach that is based on the plumbline technique of the previous section. Consider that the camera acquires two images of a textured plane with otherwise unknown structure. We suppose that we can match the two images densely; the matching does not actually need to be perfectly dense, but assuming it simplifies the following explanations. This is discussed below in more details.

We now describe how dense matches between two images of a planar scene allow the generation of line images and hence to apply the plumbline technique. Consider any radial line (line going through the distortion center) in the first image; the projection rays associated with the points on that line are necessarily coplanar according to our camera model. Therefore, the scene points that are observed along that radial line must be collinear: they lie on the intersection of the plane of projection rays, with the scene plane. Due to the dense matching, we know the projections of these collinear scene points in the second image. By considering dense matches of points along n radial lines in one image, we thus obtain n line images in the other image, and vice versa. In addition, these line images usually extend across a large part of the image, bringing about strong constraints.

We now simply stack all plumbline constraints (4) for all pairs of images, and solve for the distortion parameters as in §4. Here, we have assumed the knowledge of the distortion center (in order to define radial lines); the distortion center can of course also be estimated, using e.g. the exhaustive approach of §4.2. Moreover, the input, once rectified, can be given to a classical plane-based self-calibration algorithm to obtain a full metric calibration, using e.g. [21].

Dense Matching. Dense matching can be achieved rather straightforwardly. If the camera acquires a continuous image sequence, most existing optical flow algorithms can be applied for successive frames and their results propagated in order to obtain a dense matching between two images with a substantial motion between them. In addition, the fact that a planar scene is observed eliminates the occlusion problem. If the scene is not sufficiently textured, but only allows to extract and track sparse interest points, then we proceed as follows. We extract dominant lines in each image using a Hough transform of the extracted interest points, and only keep the lines passing near the current distortion center estimate. These are almost radial lines. An example is shown in fig. 3a,b. The rest of the self-calibration is as above.

Constrained Camera Motions. Another way to obtain line images without the need for linear features in the scene is to acquire images under constrained

camera motions. A first possibility is to carry out pure rotations about the optical center, as suggested also by [19]. The scene can then be assimilated to a plane, and the above self-calibration method can be directly applied. A second possibility is to perform pure translations (with e.g. a tripod) and to track image points across several images. In this case, any point track constitutes a line image (an example is shown in fig. 3c,d).
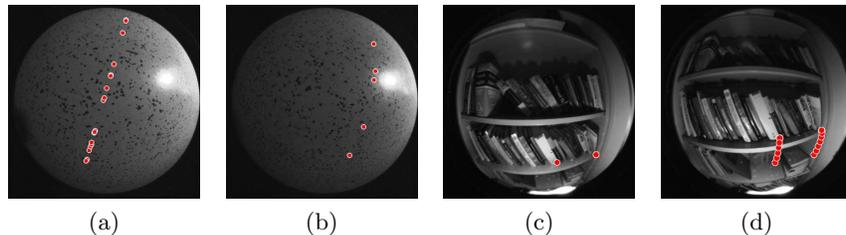

(a) (b) (c) (d)

Fig. 3. (a)+(b) Two images of a planar scene. a) shows interest points lying on a radial line in the first image and b) corresponding points in the second image. **(c)+(d)** Two images of a general scene, taken with pure translation. c) shows two interest points in the first image and d) their paths, accumulated in the last image.

6. Results and Analysis

We tested our algorithm with data acquired from real and simulated cameras. An 8.0 mm lens, a 3.5mm fisheye lens and a para-catadioptric camera were used. We also simulated ten cameras featuring distortions from small to very large.

6.1. Convergence Analysis of the Distortion Center Detection

Two aspects of convergence of the plumbline method were evaluated. First, evaluating if the minimization of the constraints given by (6) instead of (5) leads to similar results. This is not critical though, as the path of the optimizer needs not be the same to ensure convergence. On the other hand, if the paths are similar, it suggests that the convergence pace is not penalized too much with the simplified cost function. We proceeded as follows. For samples of distortion center positions in a box around the initial position, we computed the two cost functions and found their minima (fig. 4a,b). We see that the functions' general shapes are almost identical, as well the positions of their respective minima. Another evaluation consists in initializing the distortion center randomly around the optimal one and finding the minima of the two cost functions. Figure 4c shows the average distance between these minima, as a function of the distance of the given distortion center from the optimal one. It is generally small, suggesting that both cost functions may lead to similar optimization paths.

Secondly, the overall convergence was tested with simulated and real data. In the first case, three criteria were considered: the number of line images given
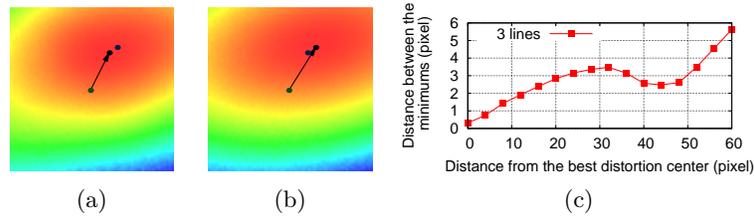
(a)   (b)   (c)

Fig. 4. Plots of cost functions and optimization paths associated with **(a)** eq. (5) and **(b)** eq. (6). **(c)** Distance between minima of these two cost functions, with respect to distance of current estimate of distortion center from optimal one. Data from the 3.5mm fisheye lens.
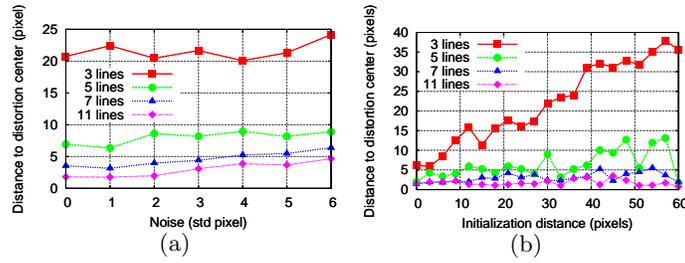

(a)   (b)

Fig. 5. Precision of the recovered distortion center on simulated data w.r.t. **a)** noise and number of lines, **b)** number of lines and initialization distance.

as input, the amount of noise added to the data and the distance of the given initial distortion center from the true one. For each simulated camera, up to 11 line segments were generated randomly, Gaussian noise of standard deviation 0 to 6 pixels was added to image point coordinates and these were then quantized to pixel precision. For every camera, 50 initial values for the distortion center were randomly chosen in a circle of 60 pixels radius around the true position (for images of size 1000×1000) and given as input to the algorithm. This a realistic test considering that for our real cameras, we found that the estimated distortion center converged to around 30 pixels from the initial value (image center) in the worst case. The results in fig. 5 show that the number of lines has a much larger impact on the quality of the recovered distortion center than the noise and the initialization distance. This is especially true when the number of line is larger than 7.

6.2. Plumbline Calibration

We acquired images of lines with our real cameras, calibrated the distortion and then performed rectification. Once again, we tested the convergence and also the quality of the rectification by checking the collinearity of rectified line images. Convergence was never found to be an issue, especially for the two dioptric lenses (fig. 6). Even with a really bad initialization of the distortion center, resulting in a poor initial estimate of the distortion function, the algorithm converged
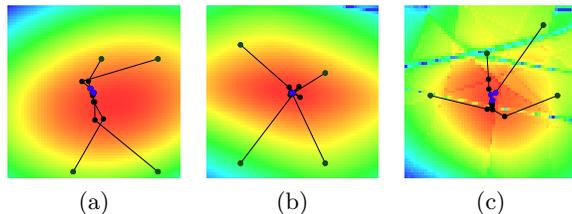
(a)   (b)   (c)

Fig. 6. Convergence examples of the algorithm for **a)** the 8.0 mm, **b)** the 3.5 mm fisheye, **c)** the para-catadioptric. The density plots show the value of the cost function explained at the end of §4.2, with f computed using distortion center positions (c_x, c_y) in a box of 60×60 pixels around the final distortion centers. In dark-green, different initializations of the algorithm; in black, the centers at each step of the algorithm; in purple, the final centers.
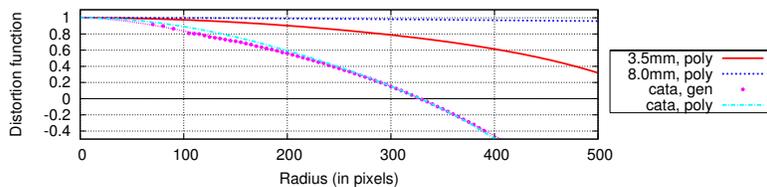

Fig. 7. Calibrated distortion functions for our real cameras. *poly* refers to (1) and *gen* to (2). For the 8.0 and 3.5mm, both representations lead to virtually identical results (details at table 1).

surprisingly fast (fig. 8). The distortion functions for our real cameras are shown in fig. 7 as well as rectified images in fig. 9 (images not used for the calibration). We compared our approach with the one presented in [6], run on the same data. Since that approach performs non-linear optimization, it can easily incorporate different distortion models. Results for different models are shown in table 1; we initialized the distortion centers with the one that was estimated with our approach and the distortion function as a constant.

Details are given in fig. 10 for the catadioptric cameras. We observe that a polynomial function did not give satisfying results. Using higher degrees (up to 10) and changing the distortion function did not give much better results. On the other hand, we see that a division function is very well suited to model the distortion in the image.

6.3. Self-Calibration from Real Sequences

Two sequences were tested. In the first one, points were tracked from a flat surface (our laboratory floor) with a hand-held camera. In the second case, a tripod was used and the camera was translated in constant direction. Overall, the results were satisfying although not as precise as with the direct plumbline technique using images of actual linear features. Results are summarized in table 2; values shown were computed like explained in table 1 and using images of
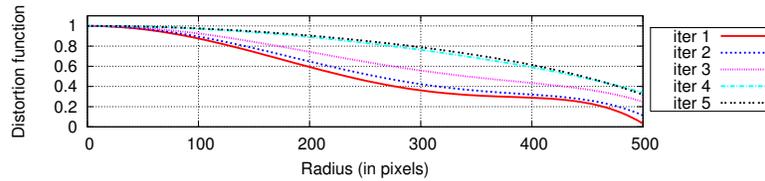
Fig. 8. The distortion function of the fisheye lens, at different iterations of the calibration algorithm for an initial center very far from the true position (200,400). The final estimate of (512,523) was found in only 5 iterations (image of size 1000×1000 pixels). Subsequent steps were only minor improvements.

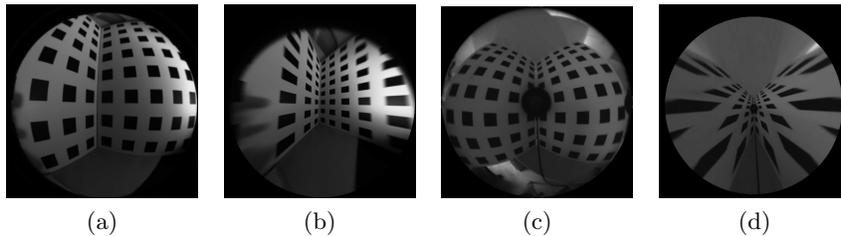|			
(a)	(b)	(c)	(d)

Fig. 9. Rectification examples. **a,b)** A 3.5mm fisheye original and rectified images. **c,d)** a catadioptric image. The radius of the principal distortion circle was estimated as 329 pixels, so circles of radius 0 to 320 pixels were rectified.

actual lines. The distortion center detection was also not as precise. The algorithm converged as usual, but not exactly to the best distortion center. In fact, it was much closer to the image center. This is explained by the fact that towards the image border, features are much more difficult to track: they are smaller and blurry. In this case, they are usually dropped by the tracking algorithm resulting in less data for large radiuses, where the distortion is the worst. Consequently, the distortion is a little bit under-evaluated and the distortion center less well constrained.

7. Conclusion

We presented flexible calibration methods for a general model for radial distortion, one plumbline type method and one for plane-based self-calibration. The methods were applied for simulated and real images of different cameras (fisheye and catadioptric). Results are satisfying, in terms of convergence basin and speed, precision as well as accuracy.

The most closely related works are [19, 20]. There, elegant though rather more involved procedures are proposed. These start with an even more general camera model than here, that does not enforce radial symmetry; only after computing and exploiting multi-view relations for that model, radial symmetry is enforced in order to compute distortion parameters. Our methods are much simpler to implement, use radial symmetry directly and can work with fewer images (two for plane-based self-calibration). Future work will mainly concern improving the

Table 1. Results using our models and algorithm (first two rows) and other models and the non-linear algorithm of [6]. Shown values refer to residual distances for fitting lines to rectified points (average and worst case). The rectified images were scaled to have the same size as the original. For the catadioptric camera, our approach used all the points, whereas the others used only the points corresponding to forward viewing cones (they failed otherwise). "—" means the algorithm did not converge without careful initialization or gave very bad results.

Models and rectifying equations	8mm		3.5mm		catadioptric	
Discrete model of (2)	0.16	1.03	**0.35**	**3.7**	**0.51**	**7.6**
Model of (1) with $d = 6$	0.16	1.12	0.35	5.5	**0.47**	**6.3**
6th order polynomial $\mathbf{p}(1 + \lambda_1\|\mathbf{p}\| + ... + \lambda_6\|\mathbf{p}\|^6)$	0.16	1.08	0.42	7.0	1.5	14.4
6th order division (non-linear)	0.16	1.08	0.36	5.6	—	—
FOV-model [6]: $\mathbf{p}\frac{\tan(\omega\|\mathbf{p}\|)}{2\tan\left(\frac{\omega}{2}\right)\|\mathbf{p}\|}$	0.23	4.86	0.54	7.9	—	—
FOV-model $+ 2^{\text{nd}}$ order polynomial	0.16	1.06	0.37	6.1	—	—

Table 2. Results for the 3.5mm fisheye with data from real sequences (fig. 3).

Models	plane		translation	
Discrete model of (2)	0.68	8.05	0.55	7.0
Model of (1) with $d = 6$	0.58	9.7	0.85	14.6

tracking for the self-calibration method and investigating the optimization of reprojection based cost functions.

References

1. S. Baker, S.K. Nayar. A Theory of Single-Viewpoint Catadioptric Image Formation. IJCV, 35(2), 1–22, 1999.
2. J.P. Barreto, K. Daniilidis. Unifying image plane liftings for central catadioptric and dioptric cameras. OMNIVIS 2004.
3. M. Born and E. Wolf. Principles of Optics, Pergamon Press, 1965.
4. D.C. Brown. Close-Range Camera Calibration. Photogrammetric Engineering, 37(8), 855-866, 1971.
5. D. Claus, A.W. Fitzgibbon. Rational Function Model for Fish-eye Lens Distortion CVPR 2005.
6. F. Devernay, O. Faugeras. Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments. MVA 2001.
7. A.W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. CVPR 2001.
8. M.M. Fleck. Perspective Projection: The Wrong Imaging Model. TR 95–01, University of Iowa, 1995.
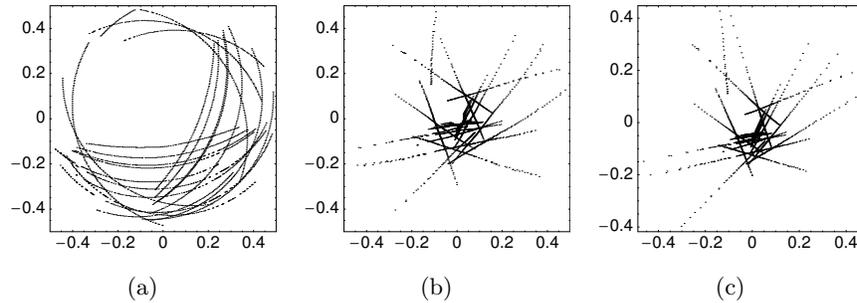9. C. Geyer, K. Daniilidis. Catadioptric Camera Calibration. ICCV 1999.

Fig. 10. Line images used as input of the algorithms and rectification results for the catadioptric camera (with principal distortion circle found at 329 pixels). **a)** Input (only shown for radius smaller than 315 pixels), **b)** Rectification with a traditional model of degree 6 (model as in third row of table 1), **c)** with the polynomial distortion function (1) and $d = 6$ (the discrete model of (2) gave almost identical results).

10. M.D. Grossberg, S.K. Nayar. A general imaging model and a method for finding its parameters. ICCV 2001.
11. R. Hartley, A. Zisserman. Multiple View Geometry in Computer Vision Cambridge University Press 2000.
12. R. I. Hartley, S. B. Kang. Parameter-free Radial Distortion Correction with Centre of Distortion Estimation. ICCV 2005.
13. Intel Open Source Computer Vision Library.
    http://www.intel.com/research/mrl/research/opencv/
14. B.Micusik, T.Pajdla. Autocalibration & 3D Reconstruction with Non-central Catadioptric Cameras. CVPR 2004.
15. S. Shah, J.K. Aggarwal. Intrinsic Parameter Calibration Procedure for A (High-Distortion) Fish-Eye Lens Camera with Distortion Model and Accuracy Estimation. Pattern Recognition, 29(11), 1775-1788, 1996.
16. D.E. Stevenson, M.M. Fleck. Nonparametric correction of distortion. TR 95–07, University of Iowa, 1995.
17. P. Sturm, S. Ramalingam. A Generic Concept for Camera Calibration. ECCV 2004.
18. R. Swaminathan, M. Grossberg, S. Nayar. Caustics of catadioptric cameras. ICCV 2001.
19. S. Thirthala, M. Pollefeys. The Radial Trifocal Tensor. A tool for calibrating the radial distortion of wide-angle cameras. CVPR 2005.
20. S. Thirthala, M. Pollefeys. Multi-View Geometry of 1D Radial Cameras and its Application to Omnidirectional Camera Calibration. to appear, ICCV 2005.
21. B. Triggs. Autocalibration from Planar Scenes. ECCV 1998.
22. X. Ying, Z. Hu Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model. ECCV 2004.
23. Z. Zhang. A Flexible New Technique for Camera Calibration. PAMI, 22(11), 1330-1334, 2000.