Nonparametric Correction of Distortion [1]

Daniel E. Stevenson Margaret M. Fleck
des@cs.uiowa.edu mfleck@cs.uiowa.edu
Department of Computer Science
University of Iowa
Iowa City, IA 52242
USA
October 1995

Abstract:

Previous work on camera calibration, and particularly work on self-calibration, has assumed that the camera is well modeled by an ideal, relatively simple, parametric model. Real camera systems, however, diverge significantly from a simple pinhole perspective imaging model. Therefore, parametric calibration models are typically extended to include parameters which model distortions from perspective. However, fitting these extra parameters greatly complicates calibration algorithms.

This paper presents a new approach to camera calibration, which avoids elaborate parametric models. The imaging geometry of a camera is modeled by a 3 degree-of-freedom stereographic imaging model, together with a smooth distortion function D. D maps the raw camera output to the ideal model, subsuming the effects of traditional parametric distortion parameters. The calibration algorithm reconstructs D from images of spheres, using the fact that spheres must appear circular in ideal stereographic projection. Accuracy and stability results for this algorithm are presented.

The remaining free parameters (image center and focal length) can be estimated using several standard methods. New methods are presented for estimating the focal length (given the image center) using a single unknown image rotation and for pasting several images, taken by rotating the camera, into a panoramic image.

Category: imaging geometry and camera calibration

[1]This research was carried out at the Department of Computer Science, University of Iowa. It was supported in part by the National Science Foundation under grants IRI-9209728, IRI-9420716, and CDA-9121985.

1 Introduction

Previous work on camera calibration, and particularly work on self-calibration, has assumed that the camera is well modeled by an ideal, relatively simple, parametric model. Camera calibration is, therefore, a process of finding the parameter values which create the best fit between the ideal model and measurements taken with the real camera. Substantial effort has been spent refining procedures for computing these parameters to high accuracy or, more often, with high repeatability, in the presence of random noise.

Real camera systems, however, diverge significantly from a simple pinhole perspective imaging model (figure 1). Therefore, parametric calibration models are typically extended to include parameters which model distortions from perspective: non-square pixels (1-2 parameters), radial distortion (1-3 parameters), and tangential distortion (1-3 parameters). However, fitting these extra parameters greatly complicates calibration algorithms. And yet more parameters may be required to accurately model the full range of variation. For example, the CCD arrays in some systems may not be perpendicular to the optical axes. Also, radial distortion functions for real lenses do not seem to be confined to a 3-parameter family of polynomials ([21, 24] and figure 1),

Finally, existing parametric calibration algorithms rely heavily on knowing the pixel location at which the optical axis intersects the image (the *image center*). A variety of methods have been proposed for locating the image center in the presence of radial distortion [1, 4, 15, 16, 17, 24]. Some algorithms [1, 23] estimate the image center as part of a larger monolithic optimization. However, it is known that this parameter is poorly determined by the image data: large changes in the position of the image center generate only small changes in image appearance [15, 24, 25]. Thus, accurate algorithms for locating the image center require very high precision measurements [1, 15].

This paper presents a new approach to camera calibration, which avoids elaborate parametric models. The imaging geometry of a camera is modeled by

1) a 3 degree-of-freedom stereographic imaging model, together with

2) a smooth distortion function D mapping the actual camera output onto this ideal model.

The function D subsumes the effects of the standard distortion parameters, as well as any other sources of distortion not described by these models.

The form of D is constrained by the fact that its output must be a stereographic image. Stereographic images of 3D are characterized by the fact that images of spheres are always circular [10, 13]. We present an algorithm that, given images of spheres, constructs a function D which will make these images near-circular. This algorithm is probably not optimal, but it is very simple and its output is sufficiently accurate for many vision

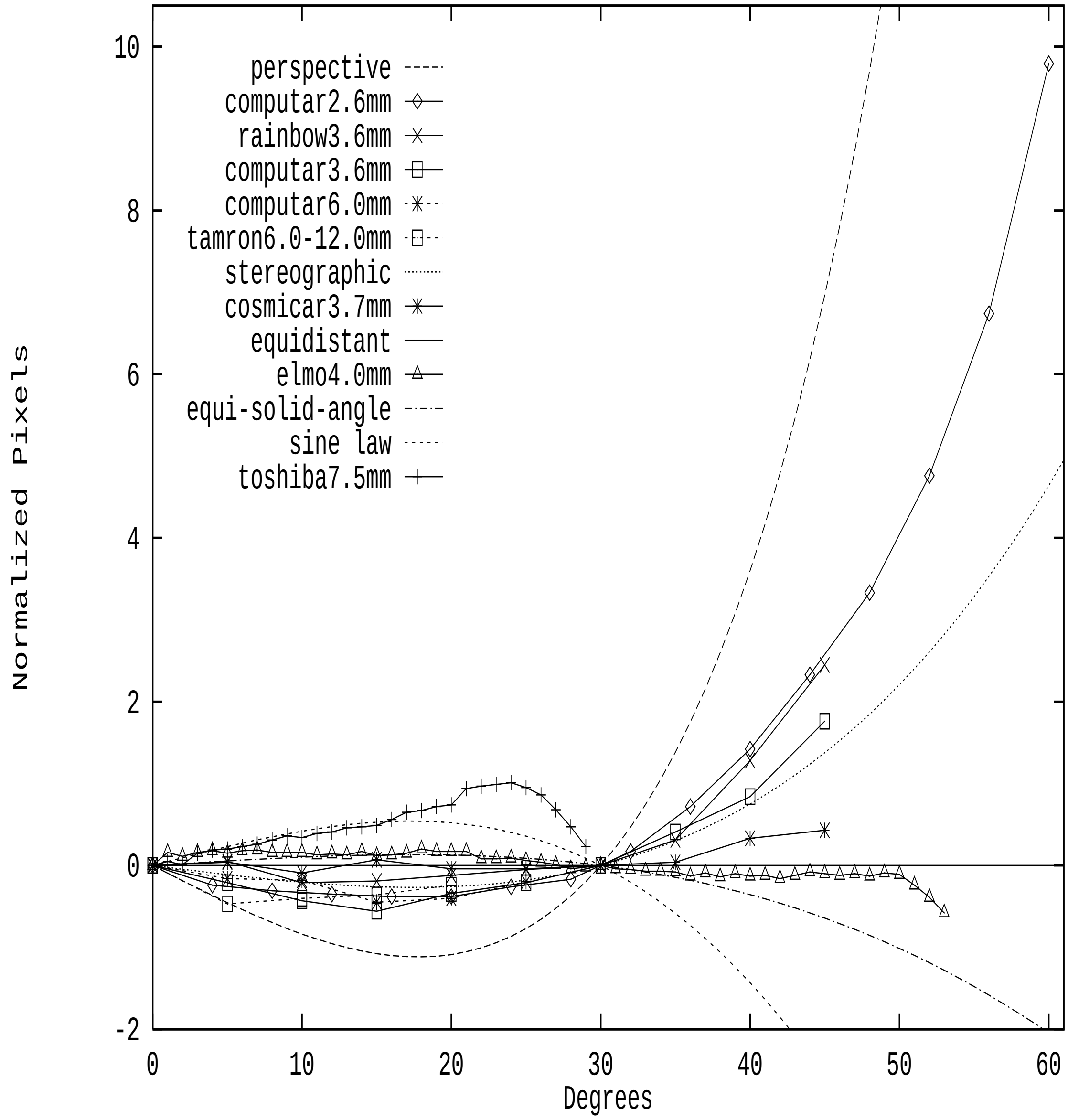

Figure 1: Radial projection functions for eight of our lenses compared to five idealized projection models. Each was normalized to have the value 30 at 30° and the line $y = x$ was subtracted. Equidistant projection lies along the horizontal axis.

applications. Furthermore, reconstruction of D does not require an estimate of the image center.

Once the image has been corrected to stereographic projection, three parameters are still unknown: the image center (two parameters) and a focal length constant. As we noted above, the image center cannot be located accurately for most camera systems and all algorithms for estimating focal length depend on an estimate of the image center. We will argue that many computer vision applications (notably object recognition) require only rough approximations to these parameters. Some applications, such as producing panoramic images by camera rotation, do not require these parameters at all. We also present a new method for estimating the focal length, given the image center, using a single unknown camera rotation.

2 Camera models

The projection geometry of most cameras can be modeled as perspective projection of the 3D world onto a sphere (the *viewing sphere*), followed by a projection π of the sphere onto a plane. For a standard ideal camera, π is rotationally symmetric about the optical axis. Five ideal *radial projection functions* are used in lens design and mathematics to map an angular distance α from the optical axis onto a distance $r(\alpha)$ from the image center:

- perspective: $r(\alpha) = k \tan \alpha$
- stereographic: $r(\alpha) = k \tan(\frac{\alpha}{2})$
- equidistant: $r(\alpha) = k\alpha$
- equi-solid angle: $r(\alpha) = k \sin(\frac{\alpha}{2})$
- sine law: $r(\alpha) = k \sin \alpha$

When used to model a digitized image, each of these ideal projections has three free parameters: the focal length constant k and the location of the image center.

Each ideal projection has useful characteristic properties. Perspective projection preserves straightness: straight lines in 3D project into great circles on the viewing sphere, and thence into lines in the planar image. Equi-solid angle projection preserves viewing sphere areas in the planar image [19]. Stereographic is conformal (preserves local shape) and projects circles on the viewing sphere (including images of 3D spheres) onto circles and lines in the planar image [10, 13]. Three of the projections support fields of view above 180°: stereographic, equidistant, and equi-solid angle. (See [7] for details and proofs.)

Perspective projection is the most popular ideal camera model. However, perspective lenses are difficult to design for wide fields of view and perspective cannot be used for fields of view wider than $120° - 140°$, due to expansion of peripheral regions (figure 2) and the related $\cos^4 \alpha$ drop-off in intensities. The other four ("fisheye") projections expand peripheral areas less and, thus, reduce intensity drop-off [7, 14, 19]. Therefore, 35mm lenses with fields of view wider than $120° - 140°$ and C-mount lenses with fields of view wider than $60° - 80°$ are designed to approximate a fisheye projection model. Lenses are commercially available with fields of view up to 220°. Details of optical designs can be found in [14, 19, 21].

It is frequently assumed that most video camera lenses are nominally perspective, and that the divergences from this ideal model are insignificant (e.g. [5]). This is not the case. Figure 1 shows the measured radial projection functions r for our eight C-mount and micro-camera lenses, ranging from a 50° (normal field of view) lens and a $30° - 60°$ zoom lens, to a 116° degree wide-angle lens. While some lenses are close to an ideal model, others lie between two ideal models. Not one of our lenses is a good approximation to the standard pinhole perspective model.

The divergence from perspective can change the apparent shape of objects as much, or more, than the elongation of peripheral regions in ideal perspective images. Two recent papers [2, 3] have proposed estimating the image center and focal length using the fact that spheres in perspective projection appear elongated towards the image center, after first pre-processing to correct radial distortion. However, images of spheres created by our lenses vary from slightly elongated along the radial lines, to slightly elongated perpendicular to them (figure 3). On such input, elongation of ellipses in the perspective image will be largely created by the algorithm correcting the radial distortion. Thus, the output of the center-finder will only be as accurate as the center estimate that was used during correction of distortion.

Based on the range of radial projection functions shown in figure 1 and in [21], it seems that the radial projection function for an uncalibrated lens may follow any reasonably-shaped curve in the range between sine-law and perspective projection. In addition, an uncalibrated camera system may have tangential distortion, due to misalignment of lens components. The CCD array may be tilted from its ideal position perpendicular to the optical axis. Finally, the ratio of horizontal to vertical distances produced by the digitizer may not be known accurately.

3 Algorithm Overview

The goal of our algorithm is to reconstruct a smooth distortion function D mapping the raw output of a camera system into a known ideal camera model. For the purposes of this paper, our ideal parametric model will be stereographic projection of the viewing sphere.

Figure 2: (Top) A 240° degree stereographic view of our lab, creating by rotating the camera into five positions. The five images were calibrated and pasted using the new nonparametric distortion algorithm. (Bottom) A 160° perspective image from the same viewpoint, created by pasting three images, using another (parametric) calibration algorithm. The gross expansion of peripheral areas makes perspective unusable for fields of view above 140°.

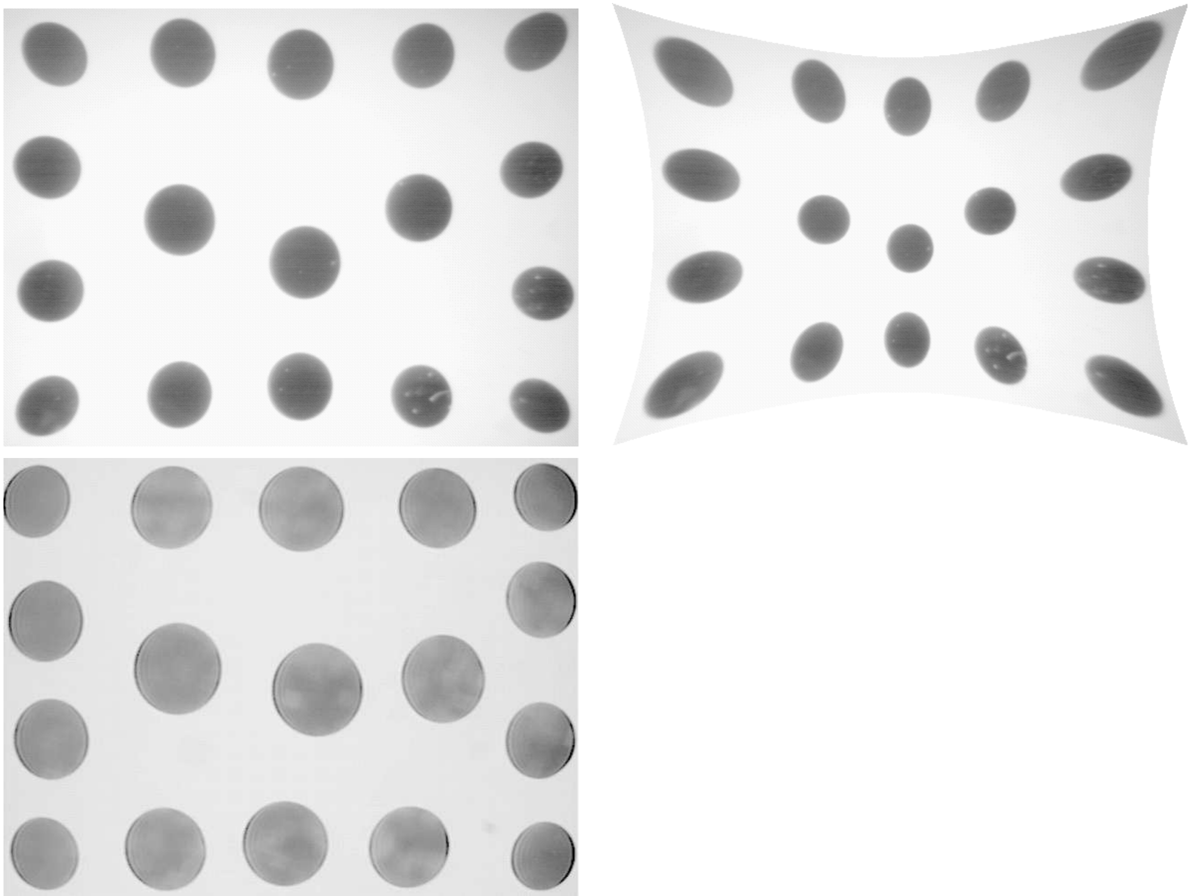

Figure 3: In an actual 116° camera image (top left), spheres are much less elongated towards the image center than in an ideal perspective image (right). In the periphery of other wide-angle images, spheres appear circular or even elongated perpendicular to the image center (bottom; field of view about 90°). Both of these camera images have been corrected so that pixels are approximately square.

Stereographic projection offers several advantages over perspective projection:

- Stereographic projection better approximates a typical C-mount lens (figure 1).
- It can represent fields of view up to about 250°, before peripheral expansion creates practical problems, whereas perspective is limited to less than 140° (figure 2).
- The images of 3D spheres are circles and, thus, the images of 3D local symmetries have 2D local symmetries [7].
- Stereographic projection maps the viewing sphere conformally onto the planar image. Therefore, an object subtending a small visual angle has the same shape (though not the same size) no matter where it appears in the field of view.

The combination of radial distortion, non-square pixels, and other types of distortion yields a wide variety of possibilities for D. If we were building a standard parametric calibration algorithm, we would approximate this set of models with a finite number of parameters. The algorithm would then determine which parameter values yield the best fit to observed data. Instead, we will specify what properties the output of D must satisfy and our algorithm will reconstruct a function whose output satisfies them.

Specifically, in ideal stereographic projection, images of 3D spheres appear circular, no matter what their size [7, 10, 13]. Our calibration algorithm is given images of spheres, taken with an uncalibrated camera. We assume that D is well approximated locally as an affine transformation, and that affine parameters vary only slowly across the image. Under these conditions, each 3D sphere appears (approximately) as an ellipse in the uncalibrated image, so long as the sphere does not subtend a very large visual angle. The calibration algorithm warps the image, so as to make these ellipses as circular as possible.

The actual implementation approximates the warping function with a piecewise affine function, as described in section 4. Figures 4 and 5 show typical output of this algorithm on images taken with our NEC NX18A CCD camera using a Computar 2.6mm lens. Most of the distortion from stereographic in these images, and in images from all our lenses, is due to the Datacube framegrabber returning images with the wrong proportions (non-square pixels). To test the algorithm on more interesting types of distortion, we simulated a pure perspective lens using calibration information obtained by the method in [24], both with square pixels and with significantly non-square pixels (aspect ratio 1.27). Figure 6 shows that the algorithm can also correct these larger deviations from stereographic projection.

Images of spheres have been used in other camera calibration algorithms, but not in this way. Penna [18] used images of spheres to correct the aspect ratio of narrow-angle perspective images. Our algorithm corrects a larger number of parameters, and works on wide-angle images. Images of spheres have been used to estimate the image center and focal length of perspective images [2, 3]. As discussed above, these methods cannot cope robustly with non-trivial radial distortion.

The plumb line algorithm for correcting an image to perspective projection [1, 22] (also used by [2, 23]) is based on a similar idea. That is, straight 3D lines should appear straight in an ideal perspective image. The algorithm examines images of many straight lines, and builds a distortion function which straightens all the lines as well as possible. However, the specific algorithm [1] used to do this correction is parametric and involves a cumbersome optimization process. More importantly, the plumb line method cannot calibrate the aspect ratio of the camera system, because the distortion created by unequal horizontal and vertical distances preserves straightness of lines. Thus, it corrects the uncalibrated image to a parametric model with four, rather than three, degrees of freedom.

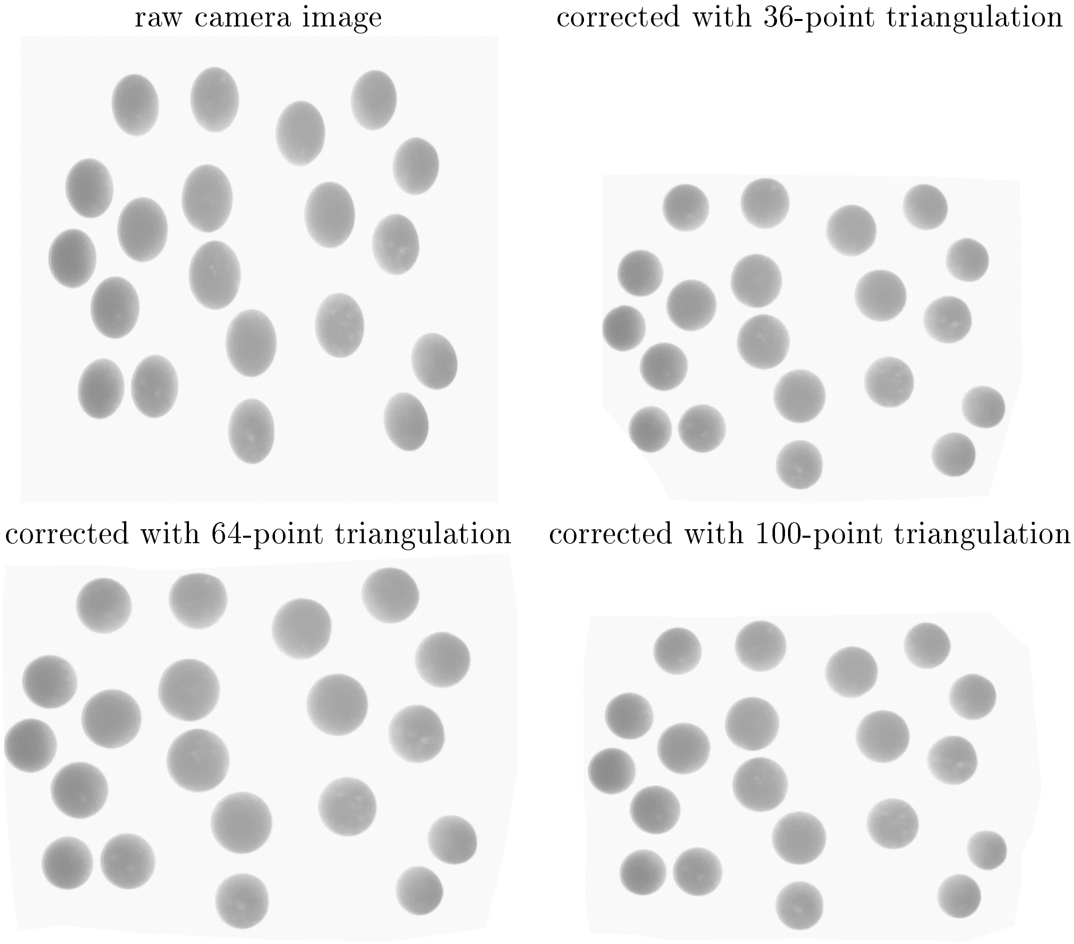

Figure 4: The raw output of our camera with a Computar 2.6mm lens (with significantly non-square pixels), and images corrected to stereographic projection using triangulations with 36, 64 and 100 points.

4 Algorithm Implementation

The input to the calibration algorithm is a set of pictures of spheres, taken with an uncalibrated camera. The algorithm first fits ellipses to all the sphere images. It selects a smaller set of ellipses, distributed more uniformly across the image and constructs a triangulation of the image, using the ellipse centers as vertices. Finally, it constructs a piecewise affine warping of this triangulation, attempting to make the image of each ellipse as circular as possible. This warping is returned as the distortion function D.

raw camera image   image corrected to stereographic

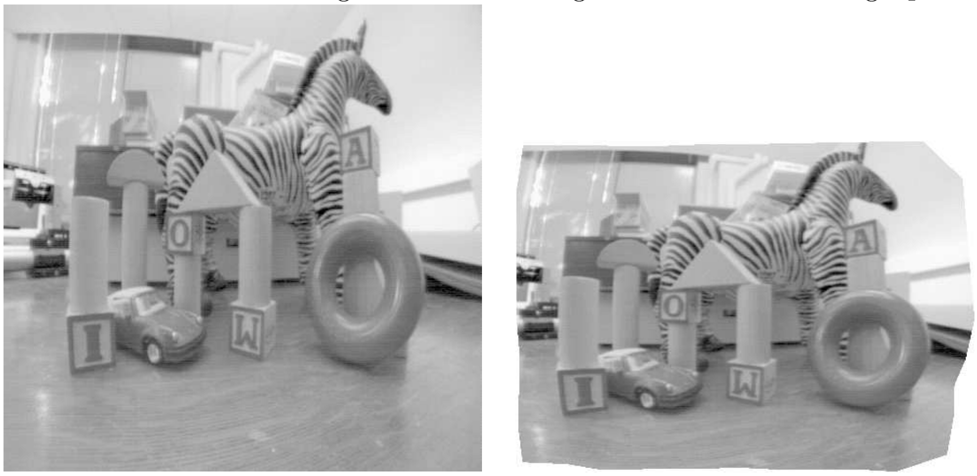

Figure 5: Raw camera output, taken with a Computar 2.6mm lens and image corrected to stereographic projection using a 100-point triangulation.

Simulated perspective, square pixels   Simulated perspective, non-square pixels

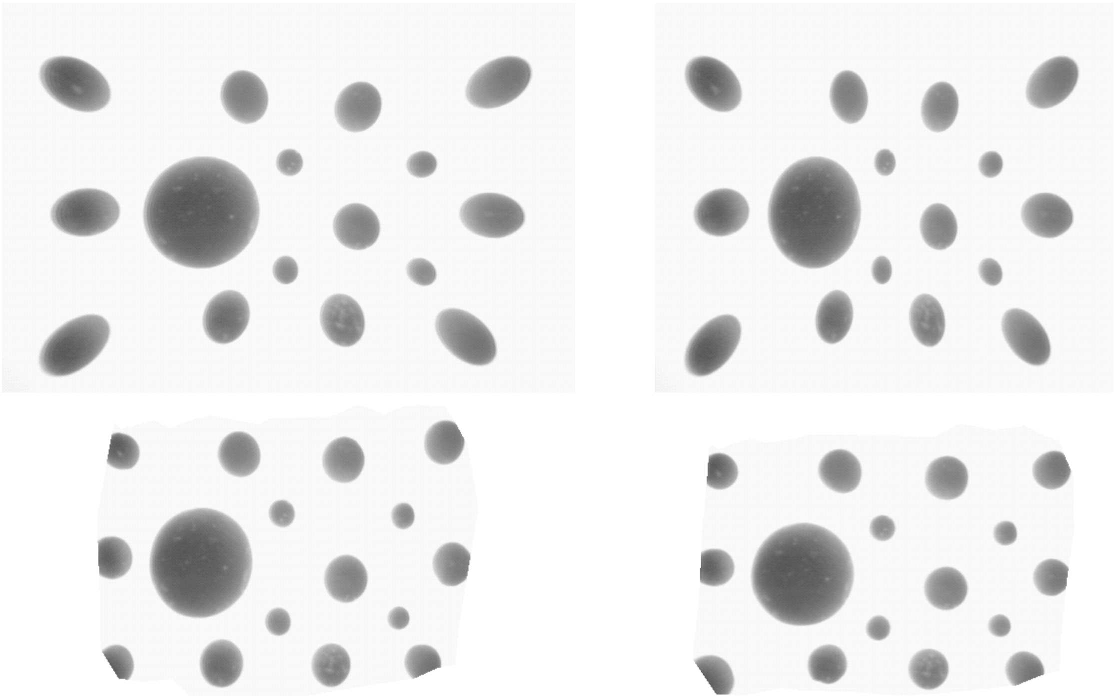

Figure 6: Top: images from two (simulated) perspective cameras, one with a correct aspect ratio and one with an incorrect aspect ratio. Bottom: images corrected to stereographic projection, using a 100-point triangulation.

4.1 Getting the ellipses

Our test spheres are wooden balls of several sizes, bought at a local craft store and spray-painted matte black. They were placed on a light table, so that their edges would be clearly defined in the images. Sphere sizes and viewing distances were arranged so that the spheres appear in the calibration images as ellipses about 15-50 pixels wide. Sphere images much wider than 50 pixels may not approximate ellipses, if the true distortion function has a complex shape. If a sphere image is significantly narrower than 15 pixels, the ellipse fitting process may become sensitive to noise.

From each image, we extract the sphere images by first thresholding to separate the sphere images from the background. Then a 4-connected component algorithm is used to separate the sphere images from each other. An ellipse is fit to each sphere image, using moments to compute the major and minor axes [11, 20]. We use a moment-based technique because it produces more stable fits than an edge-based technique [20]. We do not believe that our method requires high accuracy from the test spheres or the ellipse fitter, unless the application requires the output calibration to have high accuracy.

However, it is important that the sphere images cover as much of the image area as possible: calibration is poor for areas containing few spheres. To produce uniform coverage, we take pictures of more spheres than we require, subdivide the image into k by k rectangles, and select r sphere images from each rectangle. The center of each selected ellipse is required to be at least 15 pixels from all other selected ellipse centers.

Second, it is important to place spheres as close as possible to the edges of the image. Because the distortion parameters often change quickly near the edges of the image, significant errors are introduced if the calibration near the edges is extrapolated from points in the interior of the image. Using a more sophisticated ellipse fitter, we might be able to process ellipses which lie partly outside the boundaries of the image, and thus obtain data closer to the image edges. Alternatively, we could use smaller ellipses near the image edges.

4.2 Triangulation

We then construct a constrained Delaunay triangulation, using the ellipse centers as vertices. We use an incremental algorithm [8, 9], which must be given a polygonal region R of the plane to triangulate. If extrapolation to the extreme edges of the image is required, R should be the outer boundary of the image. However, the calibration function is significantly less accurate when extrapolation is required. Therefore, for the tests reported in this paper, we confined the triangulation to the convex hull of the ellipse centers.

We used a Delaunay triangulation because it can be computed very efficiently and because it avoids producing long, skinny triangles. It is well-known that such triangles lead to numerical instability in interpolation algorithms and they were responsible for many poor

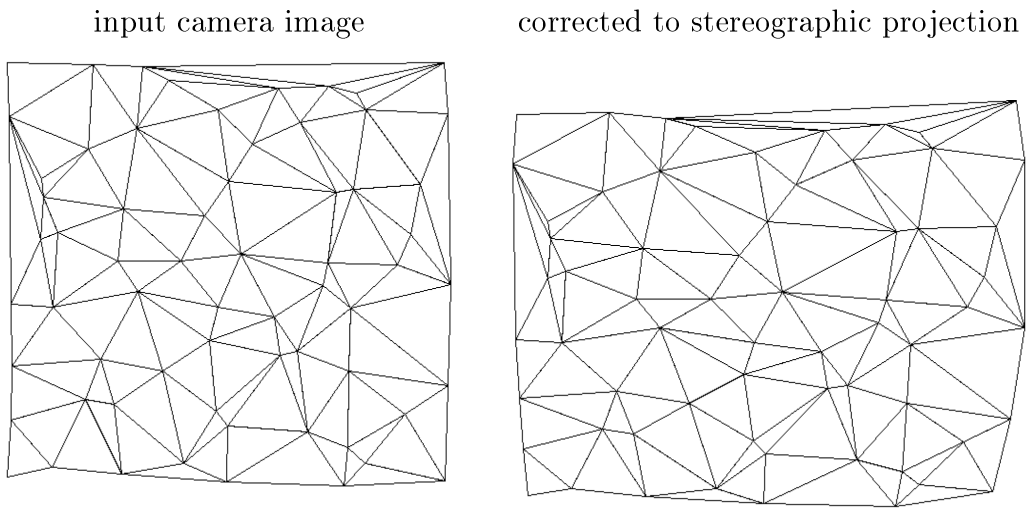

Figure 7: A typical triangulation of the convex hull of the ellipse centers, in its original form and after it has been warped into stereographic projection. The sides are nearly straight because many calibration ellipses were placed in lines near the image edges.

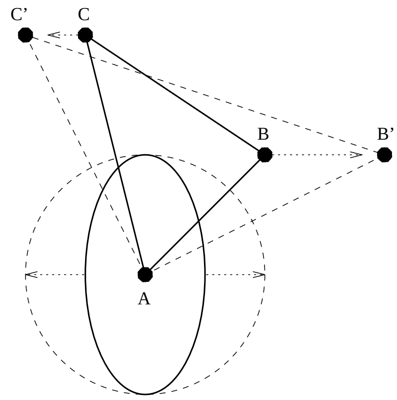

Figure 8: The ellipse centered at vertex A of a triangle is made circular by rescaling along its minor axis. This affine transformation also produces corrected locations for the other two vertices of the triangle, B and C.

results from our initial implementation of the calibration algorithm. The triangulation code does occasionally produce thin triangles adjacent to the boundary of R: we remove them after the triangulator has finished. Figure 7 shows a typical output.

4.3 Warping each triangle

We then compute the ideal output proportions for each triangle in the triangulation. These are determined by the ellipses at its vertices. An interior triangle has ellipses centered at all three vertices. Triangles on the edges of the image may have only one or two ellipses.

Consider a triangle with vertices A, B, and C and an ellipse centered at A. Our goal is to make the ellipse circular. To do this, we treat A as the origin of our coordinate system and solve for the affine transformation which stretches the image in the direction of the minor axis, so as to make the lengths of the minor and major axes equal. We then apply this affine transformation to compute the new vertex locations B' and C' (figure 8). This determines lengths for all three sides of the triangle.

Typically, a triangle will have an ellipse at more than one of its vertices. To combine estimates from several ellipses, we rescale them so that the length of one side is the same, then average the estimates for the other two sides. Finally, the lengths are converted to ratios: the desired proportions for this triangle.

4.4 Warping the triangulation

We now compute a warping for the entire triangulation. This warping must keep the topological structure of the triangulation intact. It must also achieve, as much as possible, the desired output proportions for each triangle. This problem could be phrased as one of minimizing energy (the mismatch between each triangle and its desired proportions) and some optimal algorithm constructed. However, we have found that a simple, incremental method seems to work well enough for many vision applications.

We start by choosing one edge from the triangulation, near the middle of the field of view. We place this line segment into the stereographic triangulation, without any modification. This fixes the global scale and rotation of the stereographic triangulation, while we are constructing it. After the complete triangulation has been built, it can, of course, be rescaled and rotated as desired.

Then we start adding the other vertices from the triangulation, one at a time, into the stereographic triangulation. We order the points so that, when each new point is added, at least one new triangle is created in the stereographic triangulation. When there are multiple choices for which point to add next, the algorithm works its way around the boundary of the new triangulation in a spiral fashion: a breadth-first rather than depth-first expansion of the triangulation.

Suppose first that the point we want to add, call it p, is connected to the stereographic triangulation by exactly two edges. That is, one new triangle will be added. One side of this triangle is already in the stereographic triangulation, so its length is fixed. We use the desired proportions for the triangle, computed as described in section 4.3, along with this fixed length to determine where p must be placed.

In some cases, adding the new point p may require adding more than one triangle. In such cases, we use each triangle to compute an estimate for p's new position. These estimates are then averaged together to obtain a final position for the new vertex.

5 Experimental Results

We tested two aspects of the algorithm's performance: accuracy and stability. These tests were performed using an NEC NX18A color CCD camera, a Computar 2.6mm lens, and a Datacube Digicolor framegrabber. We photographed spheres in a variety of positions, resulting in a set of 300 sphere images. From this, we constructed triangulations with three different densities of measurements: 36, 64 and 100 ellipses. The ellipses were selected as described in section 4.1, with the parameters k and r set to (a) $k = 6$ and $r = 1$, (b) $k = 8$ and $r = 1$, or (c) $k = 5$ and $r = 4$.

To measure accuracy, we tested how well the algorithm succeeded at its main goal: ensuring that 3D spheres appear circular in the output image. Three warping functions were computed, using triangulations of 36, 64, and 100 ellipses. We took photographs of 77 new spheres, ranging in radius from 18 pixels to 127 pixels, and undistorted them using each triangulation. A circle C was fit to each boundary. Finally, we computed the standard deviation of the distances from the center of C to the actual boundary.

Figure 9 shows the deviations from circularity for the raw camera output. The smallest sphere boundaries show errors dominated by digitization of boundaries. (We use a nearest-pixel edge finder.) The increasing error for the larger sphere boundaries reflects distortion from stereographic projection. In the corrected images (figure 10) errors are all near the level of digitization noise. Performance does not seem to be very sensitive to the number of ellipses used to create the triangulation: results for the 36-ellipse triangulation still look good.

To test stability, we randomly generated 20 triangulations, each based on 64 ellipses. As always, the selection algorithm in section 4.1 was used to ensure uniform spacing. In an earlier pre-test, we found that non-uniform spacing could result in substantial differences between triangulations. We selected one triangulation, whose output image was about 544 by 409 pixels, as reference. The other triangulations were normalized so as to have the same global scale and orientation as the reference.

Normalization of two triangulations was done by taking a grid of 20 by 20 uniformly spaced points in the input image and warping them to stereographic using both triangulations. Taking the middle of the field of view to be the origin, vectors from the origin to each warped point were computed for both stereographic images. The magnitude ratio and orientation difference of matching vectors (vectors created from warped images of the

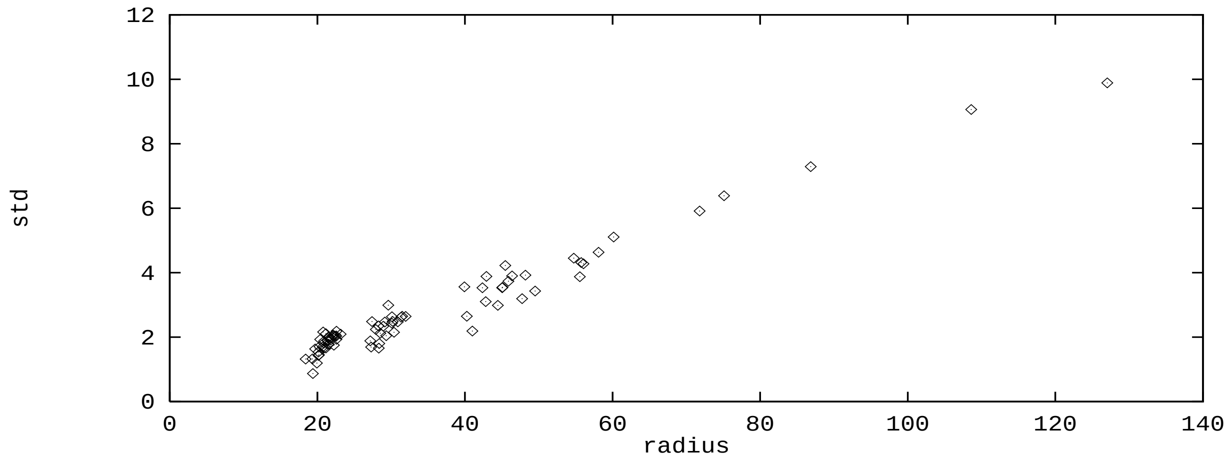

Figure 9: Deviation from circularity in the raw camera image: standard deviations of radii for each circle, plotted against the circle's radius.

same input point) were calculated. The average magnitude ratio was used to normalize the global scale and the average orientation difference was used to normalize the global orientation.

To test stability, the same grid of 20 by 20 uniformly spaced points in the input image was again warped to stereographic, but this time using the normalized triangulations. The differences between matching points were computed, for all pairs of triangulations. Figure 11 shows a histogram of these distances. The mean was only 1.84 pixels and there were relatively few extreme values.

6 Other camera parameters

After distortion has been corrected, our images are in some stereographic projection. However, we do not have values for the three stereographic projection parameters: the image center and the focal length constant k (or, equivalently, the field of view). *Whether these parameters are required and, if required, how precisely they must be determined, depends on the application and the camera's field of view.*

Without high-precision measurements (e.g. as in [15]), the image center can only be located to perhaps $\pm 5°$ [24]. However, conversely, even a ± 20 pixel displacement of the image center may cause only sub-pixel displacement of image features [15]. For applications requiring only moderate precision, e.g. object recognition, the center of the CCD array can be used as an estimate of the image center [15, 24, 25]. Alternatively, graph paper can be photographed (figure 12) and the image center estimated by intersecting the straightest

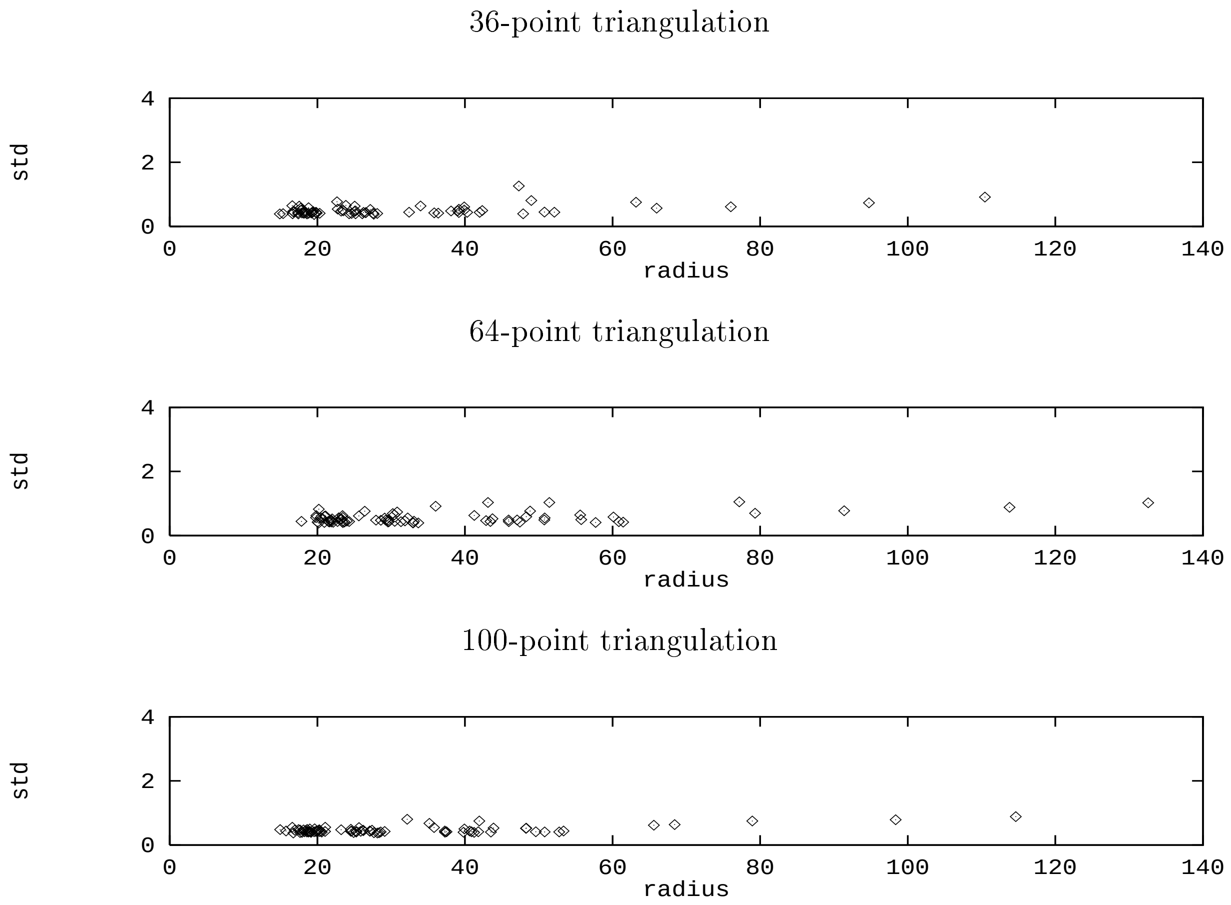

Figure 10: Deviation from circularity in the output of calibration: standard deviations of radii for each circle, plotted against the circle's radius.

horizontal and vertical lines. (Straight 3D lines appear straight in stereographic images precisely when they pass through the image center.)

Once the image center has been estimated, a variety of methods can be used to estimate the focal length constant k. These methods all seem to require an estimate for the image center but, fortunately, are not sensitive to errors in this estimate. By far the simplest method, sufficient for many purposes, is to position the camera with its lens centered on a turntable, optical axis parallel to the turntable. This can be done using the procedures in [24] or approximately by hand. One can then position the image of a distant object at the image center and observe its motion under a measured rotation.

For many applications, a rough estimate of the focal length constant (equivalently, the field of view) is sufficient. Because stereographic projection is conformal [10, 13], objects subtending small visual angles have approximately the same shape regardless of where they appear in the field of view, as illustrated in figure 5. Thus, recognition of small objects or

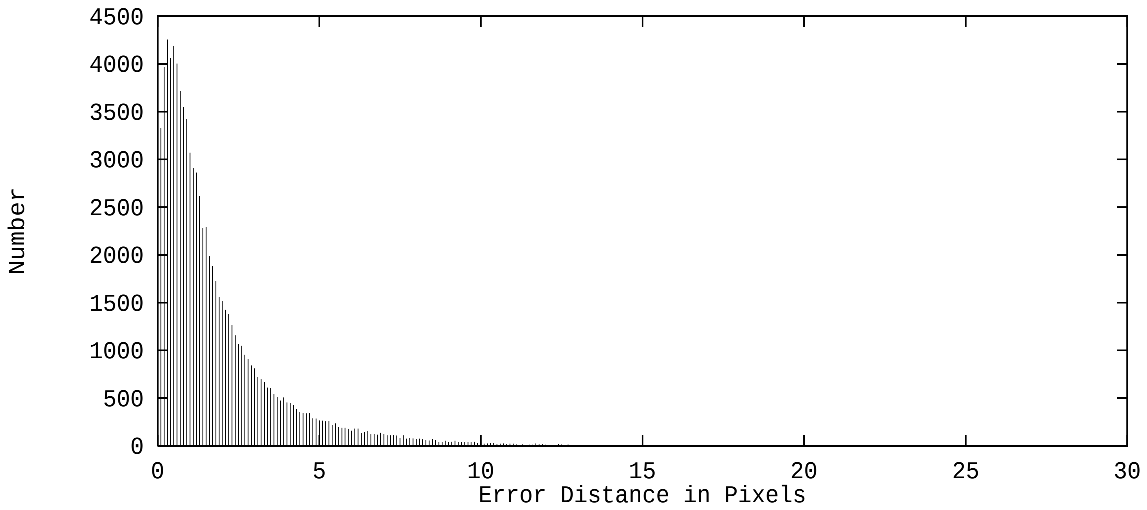

Figure 11: Distances between corresponding points in images corrected using 20 different triangulations. The reference output image was about 544 by 409 pixels. The mean is 1.84 pixels, the standard deviation is 1.98 pixels, and the maximum is 26.4 pixels. Of 87,289 pairs of points, only 76 ($< 0.1\%$) were further than 15 pixels apart.

objects not close to the camera should not require an accurate estimate of the field of view, so long as it is known not to be very wide (e.g. $\leq 120°$).

Algorithms for recognizing extended objects far from the optical axis must compensate for the fact that they are bent in stereographic images. However, as illustrated in figure 13, errors as large as 10% in the estimated field of view cause only small changes to the estimated shape. Similarly, straight 3D lines (e.g. epipolar lines in stereo matching) remain nearly straight even when reconstructed using a focal length substantially different from the correct one (figure 12).

7 Moebius transformations

Interestingly, neither the focal length nor the image center is required to create panoramic images by rotating the camera about its center of projection. Any two stereographic images, taken with fixed center of projection, are related by a Moebius transformation of the plane [10, 13]. That is, pick any arbitrary point to be the center of the image, and treat image coordinates as complex numbers. The two images are then related by a transformation of the form: $x \to \frac{\alpha x+\beta}{\gamma x+\delta}$ where α, β, γ, and δ are complex numbers, and $\alpha\delta \neq \beta\gamma$.

Suppose that we identify three pairs of corresponding points in two images: w_1, w_2, w_3,

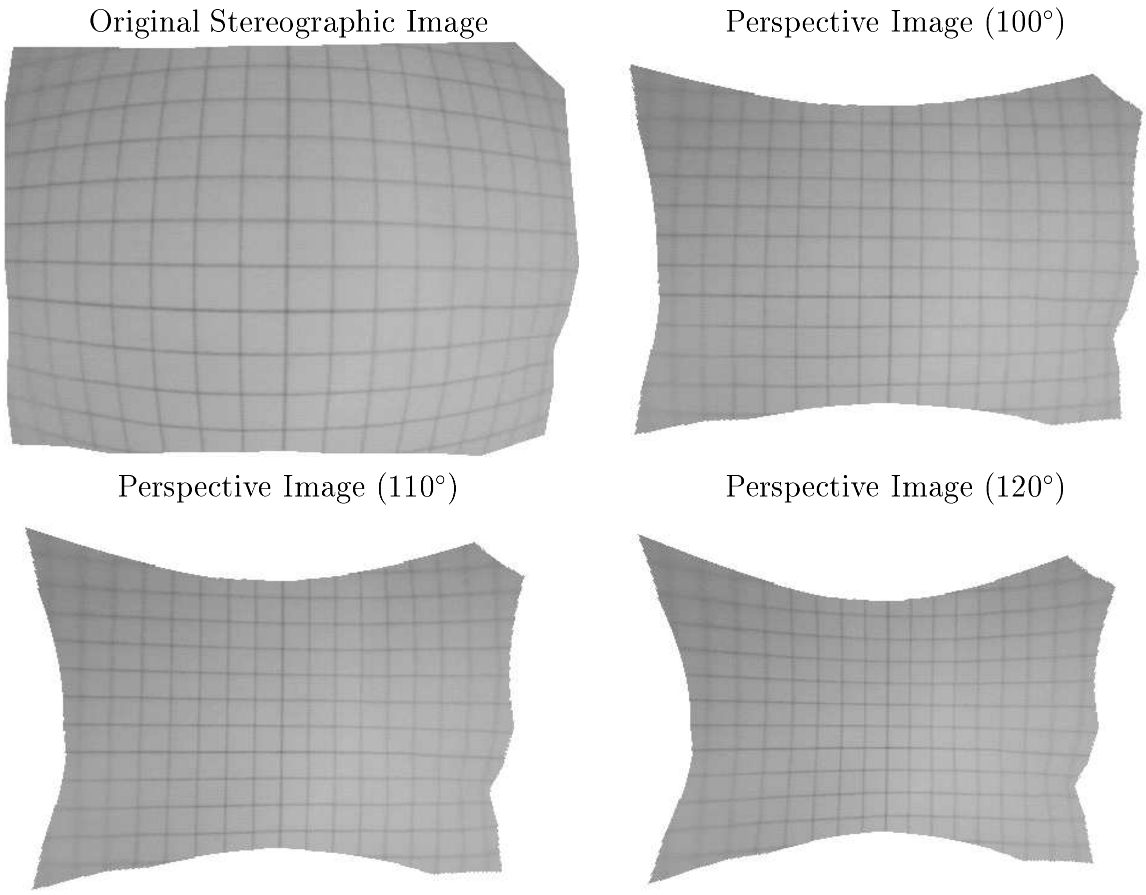

Figure 12: A piece of graph paper imaged and corrected to stereographic projection. Using the center of the original camera image as the image center and various estimates of the camera's field of view (given above each picture), the stereographic image can converted into a perspective image, in which straight 3D lines appear straight.

---

z_1, z_2, and z_3. The Moebius transformation warping the first image into the coordinate system of the second is:

$$z = \frac{(Pz_2 - Qz_3)w + (Qw_2z_3 - Pz_2w_3)}{(P - Q)w + (Qw_2 - Pw_3)}$$

where $P = \frac{z_1 - z_3}{z_1 - z_2}$ and $Q = \frac{w_1 - w_3}{w_1 - w_2}$.

We used these transformations to warp the five images making up the upper image in figure 2 into a common coordinate system. The mismatches between the images are minor and seem to be due to the fact that we marked only the minimal number of points with a low-precision mouse interface. Better feature localization and a least-squares fit to more

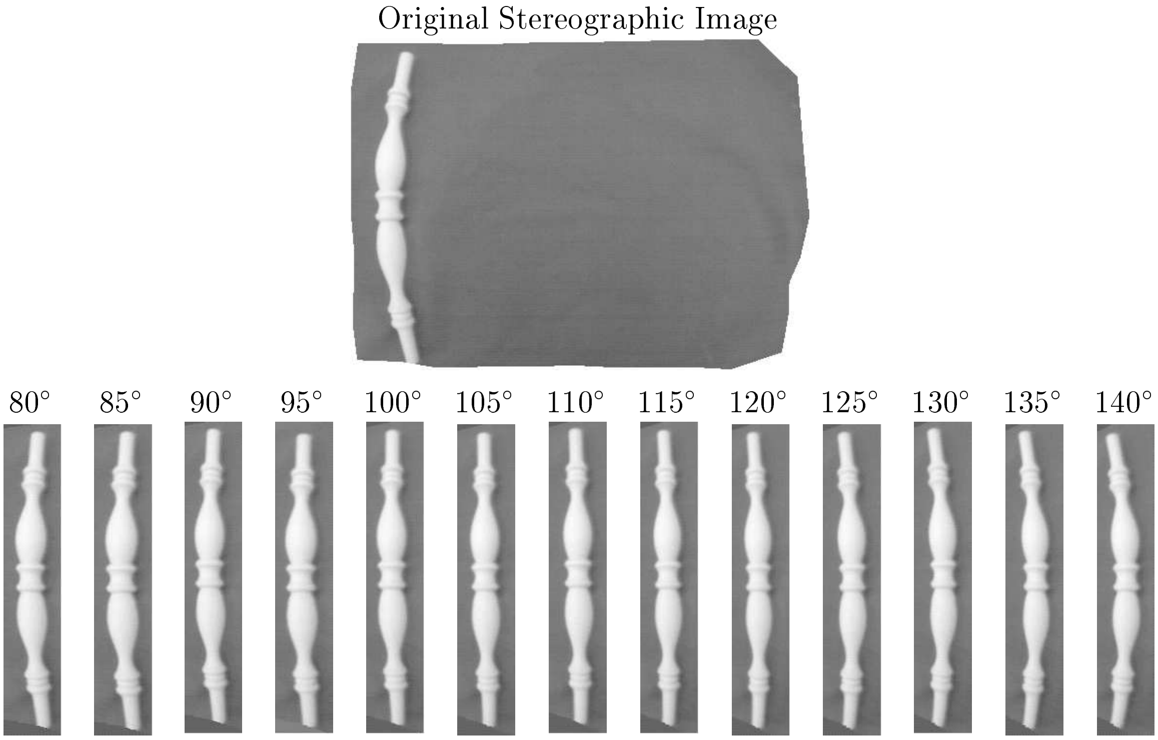

Figure 13: A long straight object appears bent when it lies in the periphery of stereographic image (top). Its true shape can be found by rotating the object into the center of the field of view (bottom). A wide range of estimates for the field of view (about 100° to 120°) produce almost the same shape.

---

points would improve accuracy.

In fact, Moebius transformations induced by camera rotations have a restricted form. Suppose that we rotate the camera about its center of projection and measure the Moebius transformation using three pairs of corresponding features, as above. Suppose also that we have an estimate of the image center. The rotation can be decomposed into a rotation by angle β through the image center, at an angle γ about the image center, plus a rotation by angle δ about the image center. The Moebius transformation has the form $x \rightarrow \frac{dx+dgkb}{-\frac{b}{gk}x+1}$ where $b = \tan(\frac{\beta}{2})$, $d = \cos\delta + i\sin\delta$, and $g = \cos\gamma + i\sin\gamma$

If the measured Moebius transformation is $x \rightarrow \frac{Ax+B}{Cx+D}$, it is straightforward to show that the focal length constant k for stereographic projection is $\sqrt{|\frac{-BD}{AC}|}$. The other parameters can be reconstructed similarly. For example, $\beta = 2\arctan(\sqrt{\frac{BC}{AD}})$.

This method works for an arbitrary, unknown rotation about the camera center, so long as β is large enough to ensure numerical stability. The general idea is similar to that used in [12] to calibrate an ideal perspective camera. However, assuming stereographic projection and an estimate for the image center greatly simplifies the equations. (The equations can be solved, as in [12], without the image center, but measurements from more than one rotation are required.)

8 Conclusions

This paper has presented a new method for calibrating images, in which a free-form distortion function is used to convert the raw camera output into an ideal stereographic image. The distortion function is estimated using the defining property of stereographic projection: images of spheres are circular. It does not require the elaborate parametric models used by previous algorithms, nor does it require a prior estimate for the image center.

Although our estimation algorithm is probably not optimal, we have shown that its output is stable and accurate enough for many computer vision applications. Furthermore, the procedure is very simple, fast, and requires only inexpensive equipment (a light table and a set of wooden balls).

We have also presented a new method for pasting together several images, taken by rotating the camera about its center of projection, using Moebius transformations. We have also introduced a new, and very simple method for estimating the focal length of a camera, given an estimate of the image center, using a single unknown image rotation.

9 Acknowledgments

The authors would like to thank Michael Covington, Wolfgang Förstner, Brian Madden, and Howard Moraff for useful pointers.

References

[1] Brown, D.C. (1971) "Close-range Camera Calibration," *Photogrammetric Engineering* 37, pp. 855–866.

[2] Beardsley, Paul, David Murray and Andrew Zisserman (1992) "Camera Calibration Using Multiple Images," *European Conf. Comp. Vision* 1992, pp. 312–320.

[3] Daucher, N., M. Dhome, and J.T. Lapresté (1994) "Camera Calibration From Spheres Images," *European Conf. Comp. Vision* 1994, pp. 449–454.

[4] Du, Fenglei and Michael Brady (1993) "Self-Calibration of the Intrinsic Parameters of Cameras for Active Vision Systems," *Proc. IEEE Conf. Comp. Vision and Pattern Recogn.* 1993, pp. 477–482.

[5] Faugeras, Olivier (1992) "Camera Self-Calibration: Theory and Experiment," *European Conf. Comp. Vision* 1992, pp. 321–334.

[6] Fiala, John C., Ronald Lumia, Karen J. Roberts, and Albert J. Wavering (1994) "TRICLOPS: A Tool for Studying Active Vision," *Intern. J. of Comp. Vision* 12/2-3, pp. 231–250.

[7] Fleck, Margaret M. (1995) "Perspective Projection: the Wrong Imaging Model," TR 95-01, Comp. Sci., U. Iowa.

[8] Floriani, Leila De and Enrico Puppo (1988) "Constrained Delaunay Triangulation for Multiresolution Surface Description," *Proceedings of the 9th International Conference on Pattern Recognition* Roma, Italy. Nov 1988. pp. 566–569.

[9] Guibas, Leonidas and Jorge Stolfi (1985) "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams," *ACM Transactions on Graphics* Vol. 4, No. 2, April 1985. pp. 74–123.

[10] Hahn, Liang-shin, *Complex Numbers and Geometry,* Mathematical Association of America, Washington D.C. 1994.

[11] Haralick, Robert M. and Linda G. Shapiro, *Computer and Robot Vision, volume I,* Addison-Wesley Publishing Co. Reading, MA. 1992.

[12] Hartley, Richard I. (1994) "Self-Calibration from Multiple Views with a Rotating Camera," *European Conf. Comp. Vision* 1994, vol. I., 471–478.

[13] Hilbert, D. and S. Cohn-Vossen, *Geometry and the Imagination,* Chelsea Publishing Co., New York, second edition, 1990, translation by P. Nemenyi of *Anschauliche Geometrie*, Springer-Verlag, Berlin, 1932.

[14] Kingslake, Rudolf, *A History of the Photographic Lens,* Academic Press, San Diego, 1989.

[15] Lenz, Reimar K. and Roger Y. Tsai (1988) "Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology," *IEEE PAMI* 10/5, pp. 713–720.

[16] Li, Mengxiang (1994) "Camera Calibration of a Head-Eye System for Active Vision," *European Conf. Comp. Vision* 1994, pp. 543–554.

[17] Oh, Sung Jun and Ernest L. Hall (1989) "Calibration of an Omnidirectional Vision Navigation System Using an Industrial Robot," *Optical Engineering* 28/9, pp. 955–962.

[18] Penna, M.A. (1991) “Camera Calibration: A quick and Easy Way to Determine the Scale Factor,” *IEEE Trans. Pattern Anal. Machine Intell.* 13, pp. 1240–1245.

[19] Ray, Sidney F. (1994) *Applied Photographic Optics,* second edition, Focal Press, Oxford.

[20] Russ, John C., *The Image Processing Handbook, 2nd ed.*, CRC Press, Roca Raton, FL. 1995.

[21] Smith, Warren J. (1992) *Modern Lens Design: A Resource Manual,* McGraw-Hill, New York.

[22] Slama, C.C. ed, *Manual of Photogrammetry*, 4th edition, American Society of Photogrammetry (1980).

[23] Stein, G.P. (1995) “Accurate Internal Camera Calibration using Rotation, with Analysis of Sources of Error,” *IEEE 5th International Conference on Computer Vision*, Cambridge MA, pp. 230–236.

[24] Stevenson, Daniel E. and Margaret M. Fleck (1995) “Robot Aerobics: Four Easy Steps to a More Flexible Calibration,” *IEEE 5th International Conference on Computer Vision*, Cambridge MA, pp. 34–39.

[25] Tsai, Roger Y., “An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision,” *IEEE CVPR* 1986, pp. 364–374.