Nonparametric Correction of Distortion

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

Abstract

Images taken with wide-angle and inexpensive medium-angle lenses show substantial distortion, which will cause many computer vision applications to malfunction. Parametric calibration algorithms cannot handle the wide range of possible distortion functions and they require a known image center, expensive equipment, and/or estimation of unneeded parameters. We present a new algorithm to remove distortion. It is similar to the plumb line method but it (a) uses images of spheres rather than lines, (b) corrects to stereographic rather than perspective projection, (c) calibrates aspect ratio, and (d) uses a nonparametric distortion model.

1 Introduction

Most computer vision applications assume that the imaging system approximates a particular ideal camera model, e.g. pinhole perspective. Real optical systems diverge from these ideal models, distorting the geometry of the output images. When these distortions are large compared to other sources of error, the application requires prior calibration. Correction of distortion is required by high-precision applications (e.g. industrial inspection). It is also required by applications that need only moderate precision but use lenses with large distortion: wide-angle lenses (e.g. surveillance, mobile robotics, TV news) or inexpensive lenses (e.g. products designed for home use, adaptive technology for the disabled). This paper considers algorithms suitable for the latter class of applications.

Images taken with inexpensive or wide-angle lenses may diverge substantially from a perspective camera model. Figure 1 shows the measured radial projection functions for our eight C-mount and micro-camera lenses, ranging from a 50° (normal field of view) lens, to a 116° degree wide-angle lens. Figure 2 illustrates the effect of this distortion on images of spheres. Not one of our lenses is a good approximation to the standard pinhole perspective model.

In addition to this radial distortion, an uncalibrated camera system may have tangential distortion (due to misalignment of lens components), the CCD array may be tilted relative to the optical axis, and the ratio of horizontal to vertical distances produced by the digitizer may not be known accurately.

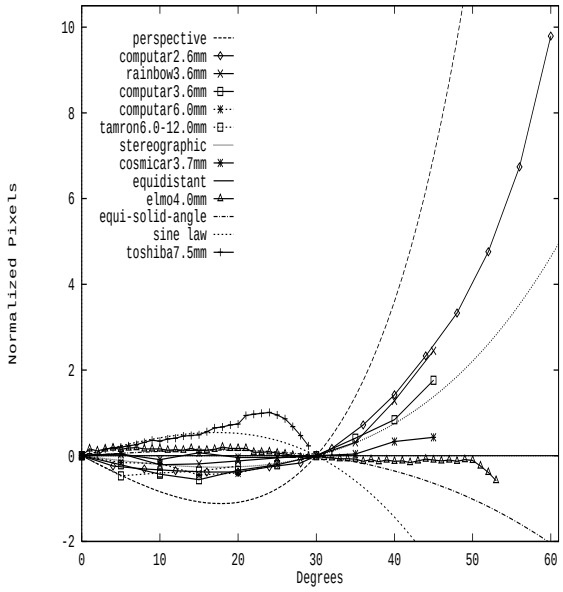

Figure 1: Radial projection functions for eight of our lenses compared to five idealized projection models. Each was normalized to have the value 30 at 30° and the line $y = x$ was subtracted.

This paper presents a new algorithm for correcting distortion which is simple, can handle large distortions, and requires no expensive equipment.

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

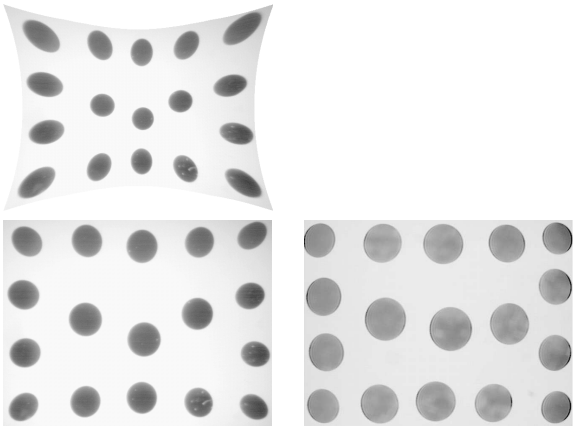

Figure 2: Top: in an ideal perspective image (116° camera image), peripheral spheres are elongated in the direction towards the image center. Bottom: in images from our lenses (corrected to have square pixels), spheres vary from less elongated (left, 116° field of view) to elongated in the wrong direction (right, 90° field of view).

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: $r(\alpha) = k \sin \alpha$

Each ideal projection has useful characteristic properties [4]. For example, perspective projection maps straight 3D lines into straight lines in the planar image. Stereographic projection projects circles on the viewing sphere (including images of 3D spheres) onto circles in the planar image [7, 8].

Although perspective projection is a popular model for narrow-angle or expensive medium-angle lenses, it is not used for wide-angle (140°–220°) or inexpensive medium-angle (over 60°) lenses. These lenses are designed to one of the other four "fisheye" projections (Figure 1, [9, 12, 14]), which support wide fields of view without the excessive expansion of peripheral regions found in perspective images (figure 3). They also reduce or eliminate the $\cos^4 \alpha$ intensity drop-off in the periphery of perspective images.

The radial projection functions for our lenses are centered about stereographic projection. We believe this is typical of C-mount lenses, because they are designed to please human viewers. Of the four fisheye projections, stereographic minimizes bending of

Figure 3: (Top) A 240° degree stereographic view of our lab, creating by rotating the camera into five positions and correcting the resulting images with the new nonparametric distortion algorithm. (Bottom) A 160° perspective image from the same viewpoint, created from three images. Gross expansion of peripheral areas makes perspective unusable for fields of view above 140°.

straight lines. Furthermore, stereographic projection is conformal, so an object subtending a small visual angle has the same shape no matter where it appears in the field of view [7, 8].

3 Parametric calibration

Most methods of calibrating distortion [2, 3, 10, 11, 15, 16, 18] estimate distortion as part of a parametric model which also includes other external and internal camera parameters. These models cannot handle the wide range of distortion patterns found in inexpensive and wide-angle images. In particular, except for [11] and [16], they use low-order models of radial distortion which are incapable of handling the full range of radial distortion functions (Figure 1 and [14]).

The parametric algorithms rely heavily on knowing the pixel location at which the optical axis intersects the image (the *image center*). This is undesirable, as the image center is poorly determined by the image data: large changes in the position of the image center generate only small changes in image appearance [16, 18]. Also, these methods require precision targets or precisely controlled camera motion.

Finally, these algorithms integrate correction of distortion from the ideal camera model with estimating the parameters of this model (e.g. focal length,

camera center). This results in unnecessary work when the application requires only rough estimates of the parameters. Or, in the case of self-calibration and perspective invariant algorithms, does not require the parameter values at all.

4 The New Approach

Two alternative methods of calibrating distortion have shown some promise. The nonparametric method of [13] models distortion as a free-form function. In theory, such a method can cope with the full range of distortions. However, their algorithm requires ground truth data that would be difficult to obtain and their results do not seem very accurate.

The plumb line algorithm for removing distortion [2] exploits the fact that 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. Once distortion has been removed, numerous methods exist for estimating the four parameters of the ideal model (image center, focal length, and aspect ratio). However, the plumb line uses a parametric distortion model, which is complex for wide-angle lenses and cannot handle complex distortion patterns.

Also, the plumb line method corrects images to perspective projection. We would prefer to correct our images to stereographic projection. Stereographic projection better approximates wide-angle lenses, so undistorting images requires less drastic interpolation. Ideal stereographic projection has only three degrees of freedom (image center and focal length). Finally, object recognition is simplified by the fact that the shape of small objects does not depend on their position in the field of view.

Our new method is similar to the plumb line method except that our calibration objects are spheres rather than lines, our output projection is stereographic, and we use a nonparametric distortion model. Although images of spheres have been used occasionally in calibration (e.g. [1]), they have been used to estimate parameters of perspective projection in low-distortion images, not to calibrate distortion.

5 Algorithm Overview

We model the imaging geometry of a camera by

1) a stereographic imaging model, and
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.

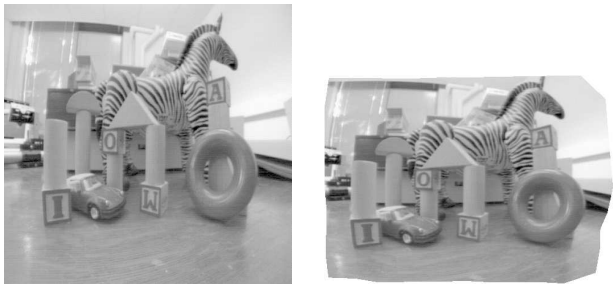

Figure 4: Raw output of a Computar 2.6mm lens (left) corrected to stereographic projection (right) using a 100-point triangulation.

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 they are the only class of images in which all 3D spheres appear circular [7, 8]. Our calibration algorithm is given images of spheres, taken with an uncalibrated camera. The algorithm then constructs a function D which will make these images as circular as possible.

We assume that D is well approximated locally as an affine transformation, and that affine parameters vary only slowly across the image. The image of each 3D sphere then approximates 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, creating a piecewise affine approximation to D.

Figure 4 shows a typical output of this algorithm on an image taken with a 116° lens. Most of the distortion from stereographic is due to our framegrabber returning images with the wrong proportions (non-square pixels). To test the algorithm on more interesting distortion, we simulated a pure perspective lens using calibration information obtained by the method in [16], both with square pixels and with significantly non-square pixels (aspect ratio 1.27). Figure 5 shows that the algorithm can also correct these larger deviations from stereographic projection.

6 Algorithm Implementation

The algorithm first fits ellipses to all the input sphere images. It selects a subset of these 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, which is returned as the distortion function D. For increased efficiency, this piecewise affine warping can be easily converted into a lookup table.

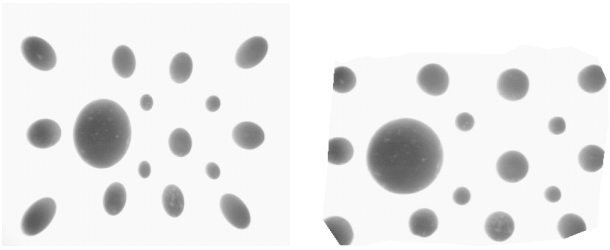

Figure 5: An image (left) from a (simulated) perspective camera, with an incorrect aspect ratio, and the same image (right) corrected to stereographic projection, using a 100-point triangulation.

6.1 Getting the ellipses

Our test spheres are wooden balls of several sizes from a local craft store and spray-painted matte black. They were placed on a light table, to create high-contrast edges. Sphere sizes and viewing distances were arranged so that the spheres appear as ellipses about 15-50 pixels wide. Much wider sphere images may not approximate ellipses if the true distortion function is complex. The ellipse fitting process may be sensitive to noise on smaller sphere images.

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. We do not believe that our method requires high accuracy from the test spheres or the ellipse fitter.

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 n 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.

6.2 Triangulation

We then construct a constrained Delaunay triangulation, using the ellipse centers as vertices. We use an incremental algorithm [5, 6], 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. Such triangles lead to numerical instability in interpolation algorithms. However, the triangulation code does occasionally produce thin triangles adjacent to the boundary of R: we remove them after triangulation.

6.3 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'. 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.

6.4 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.

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. Afterwards, it can 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 6.3, 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.

7 Experimental Results

We tested the accuracy and stability of the algorithm 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 6.1, with the parameters k and n set to (a) $k = 6$, $n = 1$, (b) $k = 8$, $n = 1$, or (c) $k = 5$, $n = 4$.

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 6 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 7) errors are all near the level of

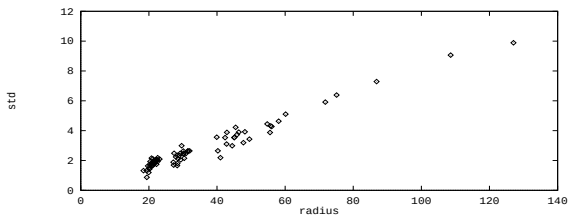

Figure 6: Deviation from circularity in the raw camera image: standard deviations of radii for each circle, plotted against the circle's radius.

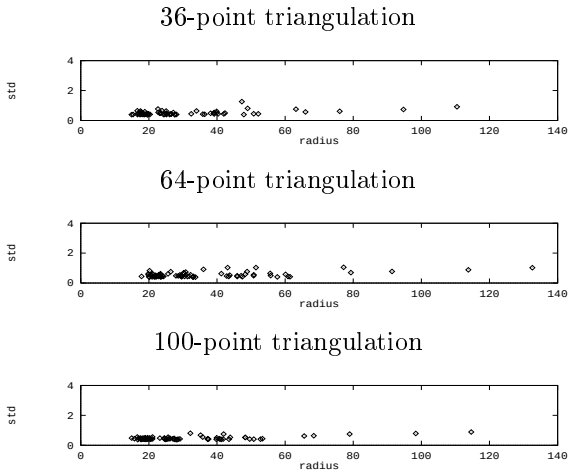

Figure 7: Deviation from circularity in the output of calibration: standard deviations of radii for each circle, plotted against the circle's radius.

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, constructed 20 different triangulations, each based on 64 randomly chosen ellipses. As always, the selection algorithm in section 6.1 was used to ensure uniform spacing. 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. See [17] for details.

Then a grid of 20 by 20 uniformly spaced points in the input image was warped to stereographic, using the normalized triangulations. The differences between matching points were computed, for all pairs of triangulations. Figure 8 shows a histogram of these distances. The mean was only 1.84 pixels and there were relatively few extreme values.

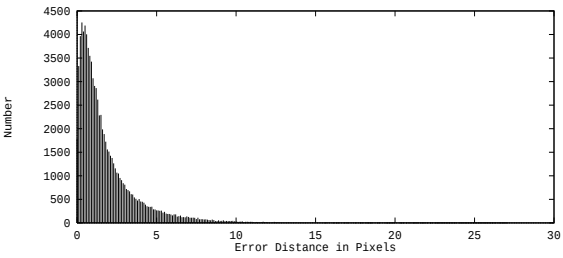

Figure 8: Distances between corresponding points in images corrected using 20 different triangulations. The reference output image was about 544 by 409 pixels. The mean is 1.84 pixels, the standard deviation is 1.98 pixels, and the maximum is 26.4 pixels. Of 87,289 pairs of points, only 76 ($< 0.1\%$) were further than 15 pixels apart.

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.

Our estimation algorithm is probably not optimal. However, 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. We hope that it will encourage more implementers to calibrate their lenses, and encourage more extensive use of calibration in applications (e.g. mobile robotics) where peripheral vision is known to be very useful.

Acknowledgments

This work was supported by NSF grants IRI-9420716 and IRI-9209728. Michael Covington, Wolfgang Förstner, Brian Madden, and Howard Moraff gave us useful pointers.

References

[1] Beardsley, Paul, David Murray and Andrew Zisserman (1992) "Camera Calibration Using Multiple Images," ECCV 1992, pp. 312–320.

[2] Brown, D.C. (1971) "Close-range Camera Calibration," *Photogramm. Eng.* 37, pp. 855–866.

[3] Du, Fenglei and Michael Brady (1993) "Self-Calibration of the Intrinsic Parameters of Cameras for Active Vision Systems," CVPR 93, pp. 477–482.

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

[5] Floriani, Leila De and Enrico Puppo (1988) "Constrained Delaunay Triangulation for Multiresolution Surface Description," ICPR 1988, pp. 566–569.

[6] Guibas, Leonidas and Jorge Stolfi (1985) "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams," *ACM Tran. on Graph.* 4/2, pp. 74–123.

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

[8] Hilbert, D. and S. Cohn-Vossen, *Geometry and the Imagination,* Chelsea Publishing Co., New York, 1990 (original publication is 1932).

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

[10] Li, Mengxiang (1994) "Camera Calibration of a Head-Eye System for Active Vision," ECCV 1994, vol. I, pp. 543–554.

[11] 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.

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

[13] Qiu, MaoLin and Song De Ma (1995) "The Nonparametric Approach for Camera Calibration," ICCV 1995, pp. 224–229.

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

[15] Stein, G.P. (1995) "Accurate Internal Camera Calibration using Rotation, with Analysis of Sources of Error," ICCV 1995, pp. 230–236.

[16] Stevenson, Daniel and Margaret Fleck (1995) "Robot Aerobics: Four Easy Steps to a More Flexible Calibration," ICCV 1995, pp. 34–39.

[17] Stevenson, Daniel and Margaret Fleck (1995) "Nonparametric Correction of Distortion," TR 95-07, Comp. Sci., U. of Iowa.

[18] Tsai, Roger Y., "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision," CVPR 1986, pp. 364–374.