

Sing Bing Kang *

Microsoft Research, Microsoft Corporation
 One Microsoft Way, Redmond WA 98052, USA
 sbkang@microsoft.com

Abstract

In this paper, we address the problem of recovering the camera radial distortion coefficients from *one* image. The approach that we propose uses a special kind of snakes called *radial distortion snakes*. Radial distortion snakes behave like conventional deformable contours, except that their behavior are globally connected via a consistent model of image radial distortion. Experiments show that radial distortion snakes are more robust and accurate than conventional snakes and manual point selection.

1 Introduction

Most cameras with wide fields of view suffer from non-linear distortion due to simplified lens construction and lens imperfection. In general, there are two forms of camera distortion, namely radial distortion and tangential distortion. In this paper, we address the problem of recovering the camera radial distortion coefficients from *one* image. We present a new technique that uses what we call *radial distortion snakes*. Unlike conventional snakes, the behavior of radial distortion snakes are globally connected via a consistent model of image radial distortion. Our radial distortion recovery is directly linked with the feature (edge) detection process.

1.1 Prior work

A lot of work on camera calibration require a calibration pattern with known exact dimensions. There are camera calibration techniques that use the scene image or images themselves, and possibly taking advantage of special structures such as straight lines, parallel straight lines, and perpendicular lines. An example is that of Becker and Bove [1]. They use the minimum vanishing point dispersion constraint to estimate both radial and decentering (or tangential) lens distortion. The user has to group parallel lines together.

* Work done while the author was with Cambridge Research Lab., Compaq Computer Corp.

Brown [3] uses a number of parallel plumb lines to compute the radial distortion parameters using an iterative gradient-descent technique. The extraction of points on the plumb lines is very manual intensive. Swaminathan and Nayar [9] use a user-guided self-calibration approach. The distortion parameters are computed from user-picked points along projections of straight lines in the image.

Stein [8] uses point correspondences between multiple views to extract radial distortion coefficients. He uses epipolar and trilinear constraints and searches for the amount of radial distortion that minimizes the errors in these constraints.

Photogrammetry methods usually rely on using known calibration points or structures [2, 3, 10]. For example, Tsai [10] uses corners of regularly spaced boxes of known dimensions for full camera calibration.

The idea of active deformable contours, or snakes, was first described in [6]. Since then, there has been numerous papers on the applications and refinement of snakes. Virtually all the snakes, some of which may be parameterized, work independently of each other. Our snakes are globally parameterized, and they deform in a globally consistent manner.

2 Finding the radial distortion parameters

We begin this section with a brief description of the lens distortion equation.

2.1 The radial distortion equation

The modeling of lens distortion can be found in [7]. In essence, there are two kinds of lens distortion, namely radial and tangential (or decentering) distortion. Each kind of distortion is represented by an infinite series, but generally, a small number is adequate.

We assume that the tangential distortion can be neglected and the principal point is the center of the

image. The radial distortion equations are then

$$x_u = x_d + x_d \sum_{l=1}^{\infty} \kappa_l R_d^l \quad (1)$$

$$y_u = y_d + y_d \sum_{l=1}^{\infty} \kappa_l R_d^l \quad (2)$$

where κ 's are the radial distortion parameters, (x_u, y_u) is the theoretical undistorted image point location, (x_d, y_d) is the measured distorted image point location, and $R_d = x_d^2 + y_d^2$.

In our approach, the user draws lines on the image that correspond to projections of straight 3-D lines. These drawn lines need not be exact. We then use snakes to search for the best-fit lines to extract radial distortion parameters. A direct approach would be to use normal snakes.

2.2 Using conventional snakes

In this method, the motion of the snakes is based on two factors: motion smoothness (due to external forces) and spatial smoothness (due to internal forces). Given the original configuration of a snake, a point on the snake \mathbf{p}_j moves by the amount $\delta \mathbf{p}_j$ at each step given by

$$\delta \mathbf{p}_j = (1 - \lambda) \sum_{k \in \mathcal{N}_j} \mu_{jk} \delta \mathbf{p}_{\text{edge},k} + \lambda \sum_{k \in \mathcal{N}'_j} \mu'_{jk} (\mathbf{p}_k - \mathbf{p}_j) \quad (3)$$

where \mathcal{N}_j and \mathcal{N}'_j are the neighborhood of pixel at \mathbf{p}_j , including \mathbf{p}_j . $\delta \mathbf{p}_{\text{edge},i}$ is the computed motion of the i th point towards the nearest detectable edge, with its magnitude being inversely proportional to its local intensity gradient. μ_{jk} and μ'_{jk} are the respective neighborhood weights. In our implementation, $\lambda = 0.5$ and $\mathcal{N}_j = \mathcal{N}'_j$, the radius of the neighborhood being 5 and the weights $\mu_{jk} = \mu'_{jk}$ being $\{1, 2, 4, 8, 16, 32, 16, 8, 4, 2, 1\}$.

Once the snakes have settled, the camera radial distortion parameters can then be recovered using a least-squares formulation.

2.3 Using radial distortion snakes

Using conventional snakes have the problem of getting stuck on wrong local minima. This problem can be reduced by imposing more structure on the snake—namely, the shape of the snake has to be consistent with the expected distortion of straight lines due to global radial distortion. For this reason, we call such snakes *radial distortion snakes*.

The complexity of the original objective function can be reduced if we consider the fact that the *effect of radial distortion is rotationally invariant about*

the principal point, ignoring asymmetric distortions due to tangential distortion and non-unit aspect ratio.

This method has the following steps:

1. For each snake, find the best fit line,
2. Rotate each snake about the principal point so that the rotated best fit line is horizontal. Let the angle of this rotation be α_i for the i th snake.
3. Estimate best fit set of radial distortion parameters $\kappa_1, \dots, \kappa_L$ from the rotated set of lines (described shortly).
4. Find the expected rotated distorted point $\mathbf{p}'_j = (x_j, y_j)$, whose undistorted version lies on a horizontal line, i.e.,

$$y_j \left[1 + \sum_{l=1}^L \kappa_l (x_j^2 + y_j^2)^l \right] = h \quad (4)$$

Given the initial points $(x_j^{(0)}, y_j^{(0)})$, we take $x_j = x_j^{(0)}$ and iteratively compute y_j from

$$y_j^{(k+1)} = \frac{h}{1 + \sum_{l=1}^L \kappa_l (x_j^2 + y_j^{(k)2})^l} \quad (5)$$

until the difference between successive values of $y_j^{(k)}$'s is negligible.

5. Update points using current best estimate of κ 's and edge normal. In other words, the point \mathbf{p}_j is updated based on

$$\mathbf{p}_j^{\text{new}} = \eta \mathbf{p}_j^{\text{kappa}} + (1 - \eta) \mathbf{p}_j^{\text{normal}} \quad (6)$$

with $0 \leq \eta \leq 1$. $\mathbf{p}_j^{\text{normal}}$ is the expected new position of the snake point using the conventional snake approach (see (3)). For the i th snake, $\mathbf{p}_j^{\text{kappa}}$ is obtained by rotating \mathbf{p}'_j (calculated from the previous step) about the principal point by $(-\alpha_i)$.

6. Iterate all the above steps until overall mean change is negligible, or for a fixed number of iterations. The latter condition is adopted in our work.

In our case, we set the time-varying function of η to be linear from 0 to 1 with respect to the preset maximum number of iterations.

To find the radial distortion parameters κ_l 's given *rotated* coordinates, we minimize the objective function

$$\mathcal{E} = \sum_{i=1}^{N_s} \sum_{j=1}^{N_{\text{pts},i}} \omega_{ij} \left[y_{ij} \left(1 + \sum_{l=1}^L \kappa_l R_{ij}^l \right) - h_i \right]^2 \quad (7)$$

where N_s is the number of snakes, $R_{ij} = x_{ij}^2 + y_{ij}^2$ and ω_{ij} is the confidence weight associated with the j th point in the i th snake. In our work, ω_{ij} is the edge strength at (x_{ij}, y_{ij}) . We use the rotated versions in order to simplify the analysis of the objective function.

Taking the partial derivatives with respect to the unknown parameters and equating them to zero, we get a linear system of equations that can be easily solved for κ_n 's. For $L > 1$, we estimate the κ 's in succession. In other words, we first estimate κ_1 , followed by κ_1 and κ_2 , and so on, up til the last stage where we estimate all the radial distortion parameters $\kappa_1, \dots, \kappa_L$.

3 Results

In this section, we present results from both synthetic and real images. For all the experiments described in this section, we recover κ_1 and κ_2 only, i.e., we set $L = 2$. This is generally sufficient for low to moderately distorted images in practice.

3.1 Experiments using synthetic images

In our first set of experiments, we use synthetic images containing straight lines and distort them with known radial distortion parameters. In addition, we vary the image noise to see how it affects both the conventional and radial distortion snake algorithms. In particular, the actual radial distortion parameters corresponding to $\kappa_1 = 10^{-6}$ and $\kappa_2 = 10^{-10}$ are applied to images with a resolution of 480×512 . The gaussian image noise (specified by the standard deviation in intensity level σ) is varied from 0 to 100 intensity levels. Figure 1 shows results for a test image with $\sigma = 100$. It is clear from Figure 1 that the radial distortion snakes yielded a better result than that of conventional snakes.

The results of the series of experiments are shown in Figure 2. As can also be seen, for low image noise levels, both snake algorithms exhibit reasonable robustness to image noise. However, the radial distortion snake algorithm is even more stable despite the presence of significant image noise, in comparison to the conventional snake algorithm.

3.2 Experiments using real images

Figure 3 illustrates a situation where the radial distortion snakes appear to have converged to a more optimal local minima than that of conventional snakes for the same snake initialization. This example shows that the radial distortion snakes are more tolerant to errors in snake initialization by the user. Our algorithm works for highly distorted images as well, as Figure 4 shows.

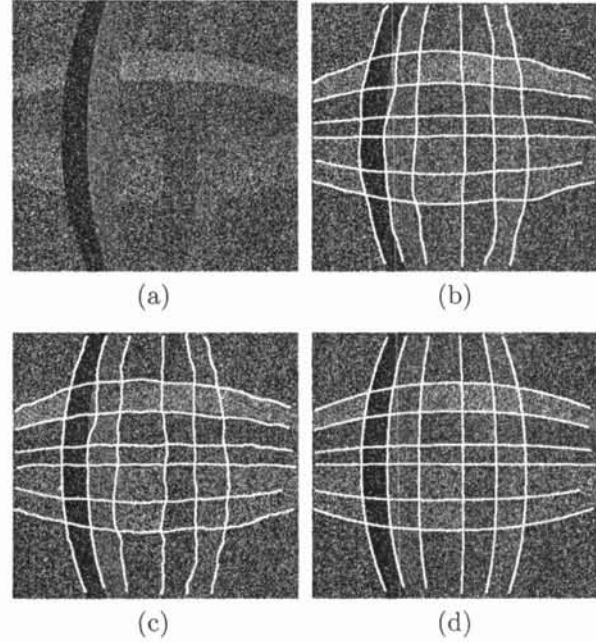


Figure 1: Synthetic image with $\sigma = 100$: (a) Original image, (b) Manually drawn lines, (c) With conventional snakes, (d) With radial distortion snakes.

Radial distortion snakes appear to have the effect of widening the range of convergence compared to conventional snakes (as exemplified by Figure 3). Despite this, wrong convergence do occur with radial distortion snakes, especially in cases of bad initial line placements.

4 Discussion

It is clear from experiments that using the radial distortion snakes is better than using conventional snakes. We have demonstrated that the radial distortion snakes find best adaptation according to best global fit to radial distortion parameters. They appear to be less prone to being trapped in bad local minima in comparison to conventional snakes. At every step, the radial distortion snakes act together to give an optimal estimate of the global radial distortion parameters and deform in a consistent manner in approaching edges in the image.

In comparison to the radial distortion snake, each conventional snake is locally adaptive and works independently of all the other snakes in the same image. They are not specialized, nor are they designed to be optimal to the task (in our case, the recovery of radial distortion parameters). This is clearly another demonstration of the benefit of incorporating global task knowledge directly in the early stages of the problem-solving algorithm. The concept of the radial distortion snake is very much in the same spirit as that of *task-oriented vision* [5].

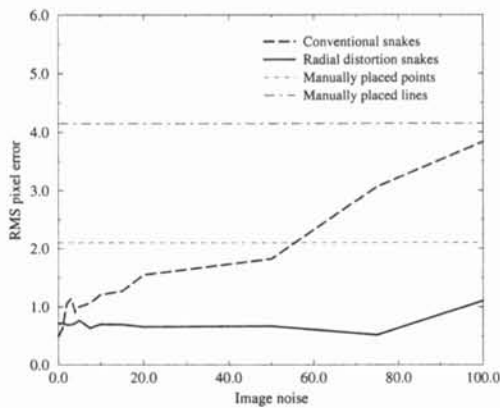


Figure 2: Graph showing the effect of gaussian image noise (standard deviation in intensity level) on RMS undistortion error E_{RMS} .

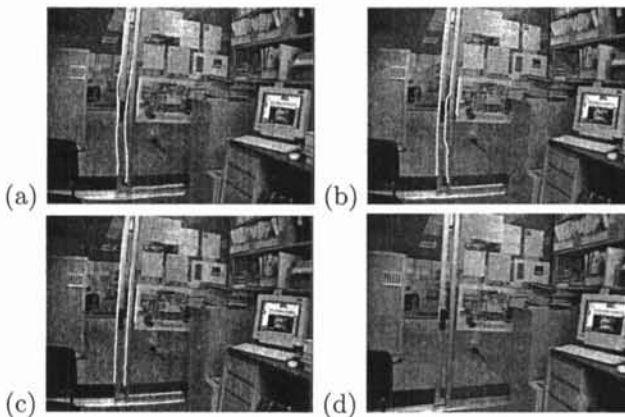


Figure 3: Office scene: (a) Initial snake configuration, and final snake configuration for (b) Conventional snakes, (c) Radial distortion snakes, (d) Corrected image from (c).

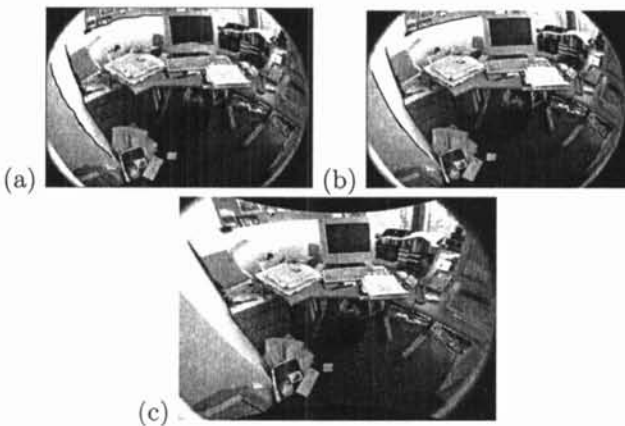


Figure 4: An example with very significant distortion: (a) Initial snake configuration, (b) Final snake configuration, and (c) Corrected image. Note that the snakes are shown in black here.

5 Summary and future work

We have described *radial distortion snakes* as a mechanism to recover radial distortion parameters from a single image. Radial distortion snakes deform in concert based on a common radial distortion model.

One direction for future work is to extend this work to estimate the principal point and tangential (or decentering) distortion parameters as well. Another area is to fully automate the process of determining radial distortion by edge detection and linking, followed by hypothesis and testing. A robust estimator may be used to reject outliers (e.g., RANSAC-like algorithm [4]).

References

- [1] S. Becker and V. B. Bove. Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In *SPIE Visual Data Exploration and Analysis II*, volume 2410, pages 447–461, Feb. 1995.
- [2] D. C. Brown. Decentering distortion of lenses. *Photogrammetric Engrg.*, 32(3):444–462, May 1966.
- [3] D. C. Brown. Close-range camera calibration. *Photogrammetric Engrg.*, 37(8):855–866, Aug. 1971.
- [4] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, June 1981.
- [5] K. Ikeuchi and M. Hebert. Task oriented vision. In *Image Understanding Workshop*, pages 497–507, Pittsburgh, PA, Sept. 1990.
- [6] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *ICCV*, pages 259–268, London, England, June 1987.
- [7] C. C. Slama, editor. *Manual of Photogrammetry*. American Soc. of Photogrammetry, Falls Church, VA, 1980.
- [8] G. P. Stein. Lens distortion calibration using point correspondences. A. I. Memo 1595, MIT, Nov. 1996.
- [9] R. Swaminathan and S. Nayar. Non-metric calibration of wide-angle lenses and polycameras. In *CVPR*, volume 2, pages 413–419, Fort Collins, CO, June 1999.
- [10] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robotics and Automation*, RA-3(4):323–344, Aug. 1987.