digital™

Semiautomatic Methods for Recovering Radial Distortion Parameters from A Single Image

Sing Bing Kang

Cambridge Research Laboratory

Cambridge Research Laboratory
Technical Report Series

CRL 97/3

May 1997

Cambridge Research Laboratory

The Cambridge Research Laboratory was founded in 1987 to advance the state of the art in both core computing and human-computer interaction, and to use the knowledge so gained to support the Company's corporate objectives. We believe this is best accomplished through interconnected pursuits in technology creation, advanced systems engineering, and business development. We are actively investigating scalable computing; mobile computing; vision-based human and scene sensing; speech interaction; computer-animated synthetic persona; intelligent information appliances; and the capture, coding, storage, indexing, retrieval, decoding, and rendering of multimedia data. We recognize and embrace a technology creation model which is characterized by three major phases:

Freedom: The life blood of the Laboratory comes from the observations and imaginations of our research staff. It is here that challenging research problems are uncovered (through discussions with customers, through interactions with others in the Corporation, through other professional interactions, through reading, and the like) or that new ideas are born. For any such problem or idea, this phase culminates in the nucleation of a project team around a well articulated central research question and the outlining of a research plan.

Focus: Once a team is formed, we aggressively pursue the creation of new technology based on the plan. This may involve direct collaboration with other technical professionals inside and outside the Corporation. This phase culminates in the demonstrable creation of new technology which may take any of a number of forms - a journal article, a technical talk, a working prototype, a patent application, or some combination of these. The research team is typically augmented with other resident professionals—engineering and business development—who work as integral members of the core team to prepare preliminary plans for how best to leverage this new knowledge, either through internal transfer of technology or through other means.

Follow-through: We actively pursue taking the best technologies to the marketplace. For those opportunities which are not immediately transferred internally and where the team has identified a significant opportunity, the business development and engineering staff will lead early-stage commercial development, often in conjunction with members of the research staff. While the value to the Corporation of taking these new ideas to the market is clear, it also has a significant positive impact on our future research work by providing the means to understand intimately the problems and opportunities in the market and to more fully exercise our ideas and concepts in real-world settings.

Throughout this process, communicating our understanding is a critical part of what we do, and participating in the larger technical community—through the publication of refereed journal articles and the presentation of our ideas at conferences–is essential. Our technical report series supports and facilitates broad and early dissemination of our work. We welcome your feedback on its effectiveness.

Robert A. Iannucci, Ph.D.
Director

Semiautomatic Methods for Recovering Radial Distortion Parameters from A Single Image

Sing Bing Kang

May 1997

Abstract

In this technical report, we address the problem of recovering the camera radial distortion coefficients from *one* image. In particular, we present three interactive methods where the user indicate lines in the single image. The first method is the most direct: the manually chosen points are used to compute the radial distortion parameters. The second and third methods allow the user to draw snakes on the image, with each approximately corresponding to a projected straight line in space. These snakes are drawn to strong edges subject to regularization. The difference between these two methods is the behavior of the snakes. In one of the last two methods, they behave like independent, conventional snakes. In the other method, their behavior are globally connected via a consistent model of image radial distortion. We call such snakes *radial distortion snakes*.

Experiments show that among the three methods, radial distortion snakes are the most robust and accurate. This is attributed to the direct link of edge fitting to the radial distortion parameters, in comparison to the other methods in which the process of locating edges is independent of radial distortion parameter estimation.

CRL Technical reports are available on the CRL's web page at
http://www.crl.research.digital.com.

Digital Equipment Corporation
Cambridge Research Laboratory
One Kendall Square, Building 700
Cambridge, Massachusetts 02139

Contents

List of Figures

List of Tables

1 Introduction

Most cameras having wide fields of view suffer from non-linear distortion due to simplified lens construction and lens imperfection. Applications that require 3-D modeling of large scenes (e.g., [1, 10, 18]) or image compositing over a large scene area (e.g., [13, 16, 17]) typically use cameras with such wide fields of view. In such instances, the camera distortion effect has to be removed by calibrating the camera and subsequently undistorting the input image.

In general, there are two forms of camera distortion, namely radial distortion and tangential distortion. In this technical report, we address the problem of recovering the camera radial distortion coefficients from *one* image. In particular, we present three interactive methods where the user indicate lines in the single image. The first method is the most direct: the manually chosen points are used to compute the radial distortion parameters. The second and third methods allow the user to draw snakes on the image, with each approximately corresponding to a projected straight line in space. These snakes are drawn to strong edges subject to regularization. The difference between these two methods is their behavior. In one of the last two methods, they behave like independent, conventional snakes. In the other method, their behavior are globally connected via a common model of image radial distortion. Such snakes are termed *radial distortion snakes*. These methods, and in particular the one that uses radial distortion snakes, are especially useful in cases where only one image is available and straight 3-D lines are known to exist in the scene.

In many cases, camera parameter recovery is disconnected from the feature detection process. While this does not present a problem if the feature detection is correct, directly linking the feature detection to parameter estimation results in more robust recovery, especially in the presence of noise. The spirit of this work is very much linked to this philosophy.

1.1 Prior work

There has been significant work done on camera calibration, but many of them require a specific calibration pattern with known exact dimensions. There is a class of work done on calibrating the camera using the scene image or images themselves, and possibly taking advantage of special structures such as straight lines, parallel straight lines, perpendicular lines, and so on. One relevant work that belong to this class 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.

Brown [3] uses a number of parallel plumb lines to compute the radial distortion parameters using an iterative gradient-descent technique involving the first order Taylor's expansion of the line fit function. The points on the plumb lines were extracted manually; the measuring process was quoted as consuming between 5-6 hours for 162 points from horizontal lines and 162 from vertical lines!

Stein [15] 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, 6, 20]. Tsai [20] uses corners of regularly spaced boxes of known dimensions for full camera calibration. In a more flexible arrangement, Faig [6] requires that the points used only be coplanar, and that there are identified horizontal and vertical points among these points. Wei and Ma [21], on the other hand, use projective invariants to recover radial distortion coefficients.

The idea of active deformable contours, or snakes, was first described in [11]. Since then, there has been numerous papers on the applications and refinement of snakes. Snakes has been applied to, for example, from tracking human facial features [22] and cells [4] to reconstruction of objects [19]. The various forms of snakes include the notion of "inflating contour" to reduce sensitivity to initialization (but at the expense of increase in parameters) [5], and the "dual active contour" [8]. In all these cases, the snakes, some of which may be parameterized, work independently of each other. In one of our proposed methods, the snakes are globally parameterized, and they deform in a globally consistent manner.

1.2 Organization

In Section 2, we briefly describe the radial distortion equation before presenting the three interactive methods of recovering the distortion parameters from one image. Subsequently, we present results using synthetic and real images in Section 3. In the same section, we propose a metric for comparing accuracy of recovered radial distortion parameters. The methods are compared and discussed in Section 4, with a summary and directions for future research given in Section 5.

2 Finding the radial distortion parameters

We begin this section with a brief description of the lens distortion equation before describing the three proposed interactive calibration methods.

2.1 The radial distortion equation

The modeling of lens distortion can be found in [14]. 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.

According to Brown [2], the lens distortion equations are

$$\begin{aligned} x_u &= x_d + \bar{x}_d \sum_{l=1}^{\infty} \kappa_l \bar{R}_d^l + [P_1(\bar{R}_d + 2\bar{x}_d^2) + 2P_2\bar{x}_d\bar{y}_d][1 + \sum_{l=1}^{\infty} P_{l+2}\bar{R}_d^l] \\ y_u &= y_d + \bar{y}_d \sum_{l=1}^{\infty} \kappa_l \bar{R}_d^l + [2P_1\bar{x}_d\bar{y}_d + P_2(\bar{R}_d + 2\bar{y}_d^2)][1 + \sum_{l=1}^{\infty} P_{l+2}\bar{R}_d^l] \end{aligned} \quad (1)$$

where κ's are the radial distortion parameters, P's are the tangential distortion parameters, (x_u, y_u) is the theoretical undistorted image point location, (x_d, y_d) is the measured distorted image point location, $\bar{x}_d = x_d - x_p$, $\bar{y}_d = y_d - y_p$, (x_p, y_p) is the principal point, and $\bar{R}_d = \bar{x}_d^2 + \bar{y}_d^2$. This model of radial distortion has been adopted by the U.S. Geological Survey [12]. In our work, we equate (x_d, y_d) to the center of the image and assume the decentering distortion coefficients to be zero, i.e., $P_1 = P_2 = ... = 0$. This leaves us with

$$\begin{aligned} x_u &= x_d + x_d \sum_{l=1}^{\infty} \kappa_l R_d^l \\ y_u &= y_d + y_d \sum_{l=1}^{\infty} \kappa_l R_d^l \end{aligned} \quad (2)$$

In our proposed calibration methods (each of which using only a single image), we expect the user to indicate the image locations of the projected straight lines in the scene. The most direct way possible is to directly on the edges and have the radial distortion parameters computed automatically.

2.2 Method 1: Using user-specified points

This method comprises two stages:

- Manual stage—the user select points on lines by clicking directly on the image,
- Automatic stage–the camera radial distortion parameters are computed using a least-squares formulation.

The radial distortion parameters are recovered by minimizing the objective function

$$\mathcal{E} = \sum_{i=1}^{N_{\text{line}}} \sum_{j=1}^{N_{\text{pts},i}} \left[(\cos\theta_i x_{ij} + \sin\theta_i y_{ij})(1 + \sum_{l=1}^{L} \kappa_l R_{ij}^l) - \rho_i \right]^2 \tag{3}$$

where (x_{ij}, y_{ij}) is the distorted input image coordinates, N_{line} is the number of lines, $N_{\text{pts},i}$ is the number of points in line i, (θ_i, ρ_i) is the parametric line representation for line i, L is the number of radial distortion parameters to the extracted, and $R_{ij} = x_{ij}^2 + y_{ij}^2$.

To get a closed form solution for the line equation, (3) can be recast as either

$$\mathcal{E}_1 = \sum_{i=1}^{N_{\text{line}}} \sum_{j=1}^{N_{\text{pts},i}} \left[y_{ij} - m_i x_{ij} - \frac{c_i}{1 + \sum_{l=1}^{L} \kappa_l R_{ij}^l} \right]^2 \tag{4}$$

for lines whose inclinations are closer to the horizontal axis, or

$$\mathcal{E}_2 = \sum_{i=1}^{N_{\text{line}}} \sum_{j=1}^{N_{\text{pts},i}} \left[x_{ij} - m_i' y_{ij} - \frac{c_i'}{1 + \sum_{l=1}^{L} \kappa_l R_{ij}^l} \right]^2 \tag{5}$$

for lines whose inclinations are closer to the vertical axis. The choice between $\mathcal{E}_1$ and $\mathcal{E}_2$ is made based on the slope of the best fit line through the set of distorted points.

Taking the partials of (4) with respect to c_n and m_n and solving yields

$$m_n = \frac{S_n(1) \sum_{j=1}^{N_{\text{pts},n}} x_{nj} y_{nj} - S_n(x) S_n(y)}{S_n(1) \sum_{j=1}^{N_{\text{pts},n}} x_{nj}^2 - S_n^2(x)} \tag{6}$$

where

$$S_n(f) = \sum_{j=1}^{N_{\text{pts},n}} \frac{f_j}{1 + \sum_{l=1}^{L} \kappa_l R_{nj}^l} \tag{7}$$

and

$$c_n = \frac{S_n(y) - m_n S_n(x)}{S_n(1)} \tag{8}$$

Similarly, for (5),

$$m_n' = \frac{S_n(1) \sum_{j=1}^{N_{\text{pts},n}} x_{nj} y_{nj} - S_n(x) S_n(y)}{S_n(1) \sum_{j=1}^{N_{\text{pts},n}} y_{nj}^2 - S_n^2(y)}, \quad c_n' = \frac{S_n(x) - m_n' S_n(y)}{S_n(1)} \tag{9}$$

The (m_n, c_n) and (m_n', c_n') representations are then converted to (θ_n, ρ_n) line representation as in (3), from which the κ_l's can directly be recovered in a similar manner to Method 3 described later (compare (3), where (θ_i, ρ_i)'s are known, with (18)). κ_l's can be found by solving the set of simultaneous linear equations

$$\sum_{l=1}^{L} \kappa_l G_{m+l} = H_m - G_m \tag{10}$$

with $m = 1, ..., L$, and

$$\begin{aligned} G_m &= \sum_{i=1}^{N_{\text{line}}} \sum_{j=1}^{N_{\text{pts},i}} g_{ij}^2 R_{ij}^m \\ H_m &= \sum_{i=1}^{N_{\text{line}}} \rho_i \sum_{j=1}^{N_{\text{pts},i}} g_{ij} R_{ij}^m \\ g_{ij} &= \cos\theta_i x_{ij} + \sin\theta_i y_{ij} \end{aligned} \tag{11}$$

For the special case of estimating only one radial distortion coefficient (i.e., $L = 1$),

$$\kappa_1 = \frac{H_1 - G_1}{G_2} \tag{12}$$

For the case of two radial distortion coefficients (i.e., $L = 2$),

$$\begin{pmatrix} \kappa_1 \\ \kappa_2 \end{pmatrix} = \begin{pmatrix} G_2 & G_3 \\ G_3 & G_4 \end{pmatrix}^{-1} \begin{pmatrix} (H_1 - G_1) \\ (H_2 - G_2) \end{pmatrix} \tag{13}$$

An alternative would be to employ the iterative gradient-descent technique involving the first order Taylor's expansion of the line fit function [3].

The problem with this approach is that it can be rather manually intensive. Typically, the extracted radial distortion parameters are more reliable with more (carefully) chosen points.

The other two methods described in the subsequent two subsection, the user draws curves over projected straight lines. The drawn curves need not be very accurate. Once this is done, these algorithms automatically reshape the snakes to stick to edges and compute the radial distortion parameters based on the final snake configuration. The difference between the these two methods is in the formulation of the snakes.

For all the subsequent snake methods described in this section, the input image is first blurred to remove high frequency changes which may prevent the snake from being attracted to a strong edge when the edge is sufficiently far away from the user specified initial snake configuration.

2.3 Method 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

$$\delta\mathbf{p}_j = (1-\lambda)\sum_{k\in\mathcal{N}_j}\mu_{jk}\delta\mathbf{p}_{\text{edge},k} + \lambda\sum_{k\in\mathcal{N}'_j}\mu'_{jk}(\mathbf{p}_k - \mathbf{p}_j) \tag{14}$$

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 ith point towards the nearest detectable edge, with its magnitude being inversely proportional to its local intensity gradient (only when it is not close to zero). μ_{jk} and μ'_{jk} are the respective neighborhood weights. λ is set according to the emphasis on motion coherency and spatial smoothness. In our implementation, $\mathcal{N}_j = \mathcal{N}'_j$, 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 are recovered using the least-squares formulation. Note that a very simplistic version of the snake is implemented here. The intent is to compare the effect of having snakes that operate independently of each other behaves worse in general than those which act in a *global* manner (as in Method 3 below), given the constraints of the types of curves that they are collectively attempt to seek.

2.4 Method 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 ith snake.

3. Estimate best fit set of radial distortion parameters $\kappa_1, ..., \kappa_L$ 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.,
$$y_j \left[1 + \sum_{l=1}^{L} \kappa_l (x_j^2 + y_j^2)^l \right] = h \tag{15}$$
Given the initial points $(x_j^{(0)}, y_j^{(0)})$, we take $x_j = x_j^{(0)}$ and iteratively compute y_j from
$$y_j^{(k+1)} = \frac{h}{1 + \sum_{l=1}^{L} \kappa_l (x_j^2 + y_j^{(k)2})^l} \tag{16}$$
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
$$\mathbf{p}_j^{\text{new}} = \eta \mathbf{p}_j^{\text{kappa}} + (1 - \eta) \mathbf{p}_j^{\text{normal}} \tag{17}$$
with $0 \leq \eta \leq 1$. $\mathbf{p}_j^{\text{normal}}$ is the expected new position of the snake point using Method 2 described earlier (see (14)). For the ith 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 (17), $\eta = 0$ is equivalent to moving the points according to the usual snake behavior as in Method 2 (function of edge normal, edge strength, smoothness, and continuity), while $\eta = 1$ is equivalent to moving to a position consistent to the current estimated radial distortion. From experiments, setting $\eta = 1$ has the effect of slower convergence, and in some cases, convergence to a wrong configuration. In general, η may be time-varying, for example, with $\eta = 0$ initially and gradually increasing to 1. n 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
$$\mathcal{E} = \sum_{i=1}^{N_{\text{s}}} \sum_{j=1}^{N_{\text{pts},i}} \omega_{ij} \left[y_{ij} \left(1 + \sum_{l=1}^{L} \kappa_l R_{ij}^l \right) - h_i \right]^2 \tag{18}$$
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 jth point in the ith snake. In our work, ω_{ij} is equated to the edge strength at image location

(x_{ij}, y_{ij}). We use the rotated versions in order to simplify the analysis of the objective function to a point where a direct closed form solution is possible.

Taking the partial derivatives with respect to the unknown parameters and equating them to zero, we get

$$\frac{\partial \mathcal{E}}{\partial h_m} = 0 \Rightarrow h_m = \frac{1}{\sum_{j=1}^{N_{\mathrm{pts},m}} \omega_{mj}} \sum_{j=1}^{N_{\mathrm{pts},m}} \omega_{mj} y_{mj} \left(1 + \sum_{l=1}^{L} \kappa_l R_{mj}^l\right) \tag{19}$$

for $m = 1, ..., N_{\mathrm{s}}$, and

$$\frac{\partial \mathcal{E}}{\partial \kappa_n} = 2 \sum_{i=1}^{N_{\mathrm{s}}} \sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij} \left[y_{ij} \left(1 + \sum_{l=1}^{L} \kappa_l R_{ij}^l\right) - h_i \right] y_{ij} R_{ij}^n = 0 \tag{20}$$

for $n = 1, ..., L$, which yields

$$\sum_{i=1}^{N_{\mathrm{s}}} \sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij} y_{ij}^2 R_{ij}^n \left(1 + \sum_{l=1}^{L} \kappa_l R_{ij}^l\right) = \sum_{i=1}^{N_{\mathrm{s}}} h_i \sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij} y_{ij} R_{ij}^n \tag{21}$$

Letting

$$A_n = \sum_{i=1}^{N_{\mathrm{s}}} \sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij} y_{ij}^2 R_{ij}^n \tag{22}$$

and rearranging the summation in the LHS of (21), we get

$$\begin{aligned} \text{LHS} &= A_n + \sum_{l=1}^{L} \kappa_l \sum_{i=1}^{N_{\mathrm{s}}} \sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij} y_{ij}^2 R_{ij}^{l+n} \\ &= A_n + \sum_{l=1}^{L} \kappa_l A_{l+n} \end{aligned} \tag{23}$$

As for the RHS of (21), substituting for h_i from (19) and changing the summation subscript in the last summation term to avoid confusion, we get

$$\begin{aligned} \text{RHS} &= \sum_{i=1}^{N_{\mathrm{s}}} \frac{1}{\sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij}} \sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij} y_{ij} \left(1 + \sum_{l=1}^{L} \kappa_l R_{ij}^l\right) \sum_{J=1}^{N_{\mathrm{pts},i}} \omega_{iJ} y_{iJ} R_{iJ}^n \\ &= \sum_{i=1}^{N_{\mathrm{s}}} \left(\bar{y}_i + \frac{1}{\sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij}} \sum_{j=1}^{N_{\mathrm{pts},i}} \sum_{l=1}^{L} \kappa_l \omega_{ij} y_{ij} R_{ij}^l\right) \sum_{J=1}^{N_{\mathrm{pts},i}} \omega_{iJ} y_{iJ} R_{iJ}^n \\ &= \sum_{i=1}^{N_{\mathrm{s}}} \left(\bar{y}_i \sum_{J=1}^{N_{\mathrm{pts},i}} \omega_{iJ} y_{iJ} R_{iJ}^n + \frac{1}{\sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij}} \sum_{l=1}^{L} \kappa_l \sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij} y_{ij} R_{ij}^l \sum_{J=1}^{N_{\mathrm{pts},i}} \omega_{iJ} y_{iJ} R_{iJ}^n \right) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{N_s} \bar{y}_i \sum_{J=1}^{N_{\mathrm{pts},i}} \omega_{iJ} y_{iJ} R_{iJ}^n + \sum_{l=1}^{L} \kappa_l \sum_{i=1}^{N_s} \frac{1}{\sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij}} \sum_{j=1}^{N_{\mathrm{pts},i}} \omega_{ij} y_{ij} R_{ij}^l \sum_{J=1}^{N_{\mathrm{pts},i}} \omega_{iJ} y_{iJ} R_{iJ}^n \\
&= C_n + \sum_{l=1}^{L} \kappa_l D_{ln} \qquad (24)
\end{aligned}
$$

for $n = 1, ..., L$.

From (23) and (24), after rearranging, we get

$$\sum_{l=1}^{L} \kappa_l (A_{l+n} - D_{ln}) = C_n - A_n \qquad (25)$$

for $n = 1, ..., L$, which is a linear system of equations that can be easily solved for κ_n's. For the special case where only one radial distortion coefficient is sufficient,

$$\kappa_1 = \frac{C_1 - A_1}{A_2 - D_{11}} \qquad (26)$$

For the case of two radial distortion coefficients,

$$\begin{pmatrix} \kappa_1 \\ \kappa_2 \end{pmatrix} = \begin{pmatrix} (A_2 - D_{11}) & (A_3 - D_{21}) \\ (A_3 - D_{12}) & (A_4 - D_{22}) \end{pmatrix}^{-1} \begin{pmatrix} (C_1 - A_1) \\ (C_2 - A_2) \end{pmatrix} \qquad (27)$$

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 κ_1, ..., κ_L.

3 Results

In this section, we present results from both synthetic and real images. The synthetic images are used to verify the accuracy of the recovered radial distortion parameters. We also artificially incorporate image noise to study the effects of noise on the snakes. Real images with significant radial distortion are used to illustrate their use in practice. 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 Metric for accuracy

In evaluating the accuracy of the recovered radial distortion parameters, it does not make much sense to compare directly the values of κ_l's for cases where $L > 1$. This is because significantly different sets of κ_l's may give rise to visually similar undistorted images. In addition, coming up with an error metric using only the actual and estimated sets of κ_l's is not trivial. In such a situation, it is more reasonable to compute an image distance-based metric that is a function of the error in undistorting the image using the actual and estimated sets of radial distortion parameters. A reasonable error measure is the RMS difference E_{RMS} between the predicted and the estimated undistorted coordinates (based on actual and estimated radial distortion parameters respectively). The error measure E_{RMS} is given by

$$\begin{aligned} E_{RMS} &= \sqrt{\frac{1}{HW}\sum_{r=-\frac{H}{2}}^{\frac{H}{2}}\sum_{c=-\frac{W}{2}}^{\frac{W}{2}}\left[\left(x_{u,rc}^{act}-x_{u,rc}^{est}\right)^2+\left(y_{u,rc}^{act}-y_{u,rc}^{est}\right)^2\right]} \\ &= \sqrt{\frac{1}{HW}\sum_{r=-\frac{H}{2}}^{\frac{H}{2}}\sum_{c=-\frac{W}{2}}^{\frac{W}{2}}[c^2+r^2]\left[\sum_{l=1}^{L}(\kappa_l^{act}-\kappa_l^{est})(c^2+r^2)^l\right]^2} \\ &= \sqrt{\frac{1}{HW}\sum_{r=-\frac{H}{2}}^{\frac{H}{2}}\sum_{c=-\frac{W}{2}}^{\frac{W}{2}}R_{d,rc}\left[\sum_{l=1}^{L}\delta\kappa_l R_{d,rc}^l\right]^2} \end{aligned} \tag{28}$$

where H and W are the image height and width respectively, $R_{d,rc} = c^2 + r^2$, and $\delta\kappa_l = \kappa_l^{act} - \kappa_l^{est}$. We use the superscripts act and est to denote actual and estimated values respectively, and subscripts u and d to denote undistorted and distorted values respectively.

3.2 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) is varied from 0 to 100 intensity levels. Figure 1 shows the results for the case with no image noise using all three methods, and Figure 2 shows results for the same test image with image noise of 100. It is clear from Figure 2 that the radial distortion snakes yielded a better result than that of conventional snakes.

The results of the series of experiments are shown in Figures 3 and 4. It can be observed from these two figures that using the RMS distortion error is a more intuitive metric to determine the degree of accuracy in recovering radial distortion parameters. As can also be seen, for low image

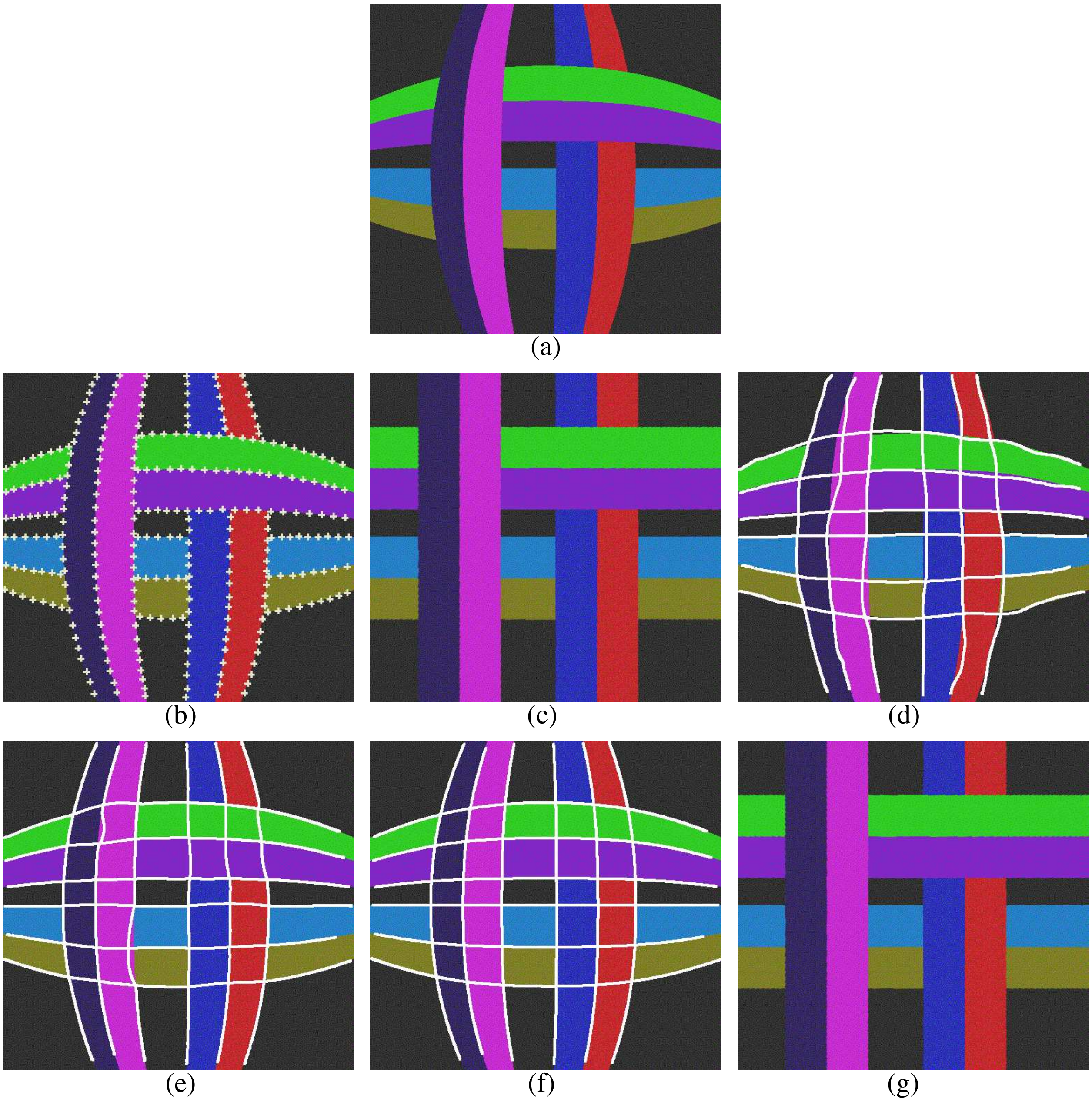


Figure 1: Synthetic image with no image noise: (a) Original image, (b) Manually selected points (299 points), (c) Undistorted image from (b), (d) Manually drawn lines, (e) After applying conventional snake algorithm, (f) After applying radial distortion snake algorithm, (g) Undistorted image from (f) (very similar to results from (e)).

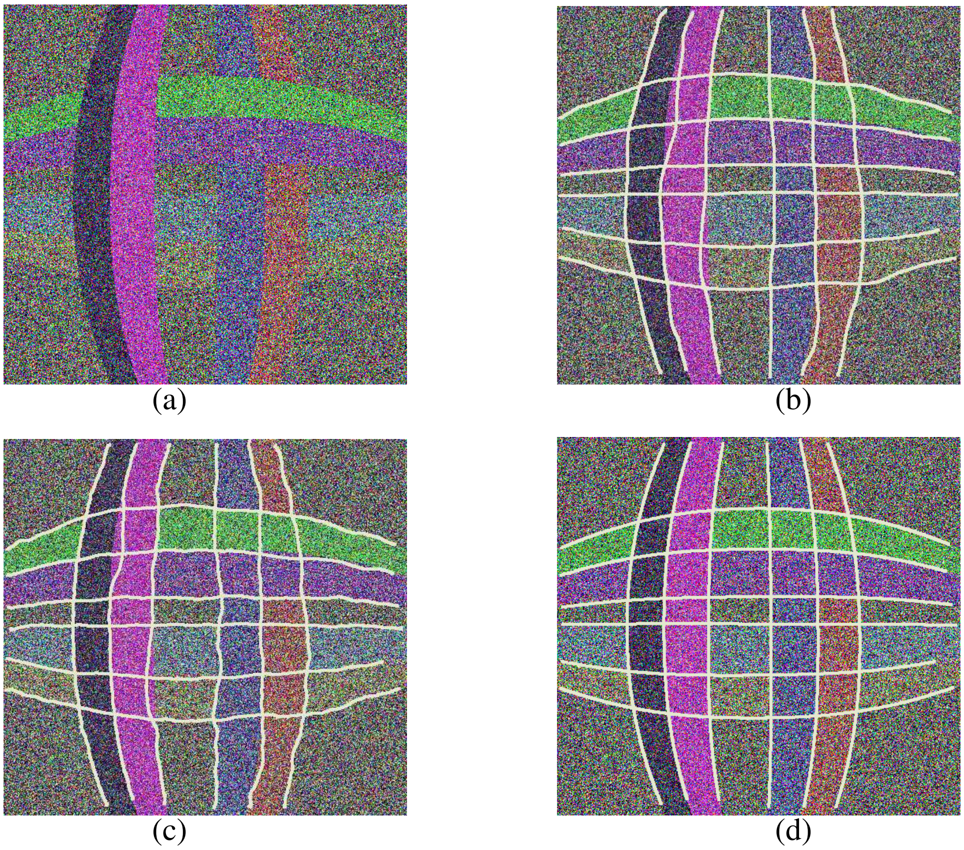
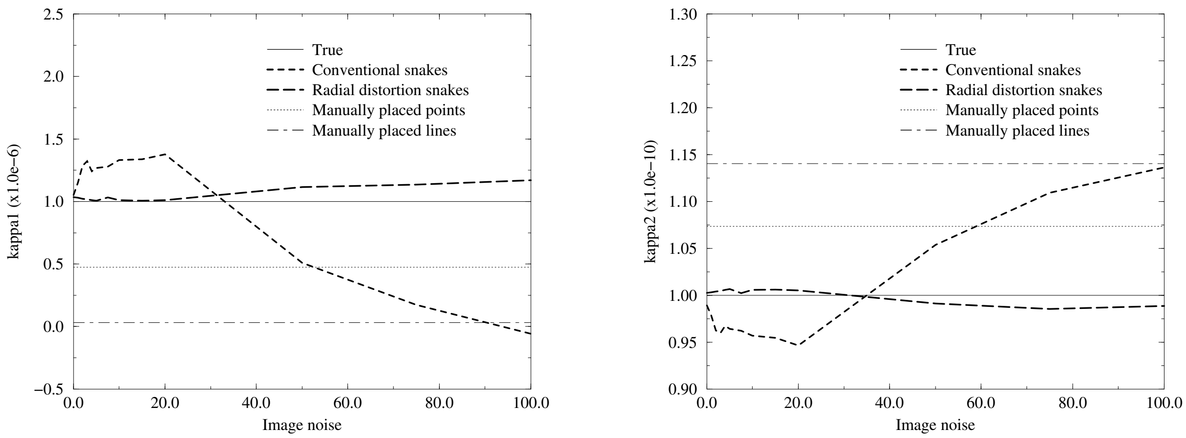


Figure 2: Synthetic image with image gaussian noise with standard deviation of 100 intensity levels: (a) Original image, (b) Manually drawn lines, (c) After applying conventional snake algorithm, (d) After applying radial distortion snake algorithm.



(a) (b)

Figure 3: Graphs showing the effect of gaussian image noise (standard deviation in intensity level) on recovered radial distortion parameters: (a) κ_1 (true value is 10^{-6}), and (b) κ_2 (true value is 10^{-10}). Note that for the manual method, the points were placed on the image with no noise. Also, for the other two methods, the snakes were initialized on the image with no noise.

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 high image noise, in comparison to the conventional snake algorithm. Note that both snake algorithms have essentially the same implementation and uses the same edge gradient estimation and same maximum number of 200 steps or iterations. The difference with the radial distortion snake is that all of the radial distortion snakes are *globally connected* via a common set of estimated radial distortion parameters. As a frame of reference, the radial distortion parameters that are estimated based directly from the user drawn lines ($\kappa_1 = 3.22 \times 10^{-8}$ and $\kappa_2 = 1.14 \times 10^{-10}$) yield an RMS pixel location error of 4.15.

3.3 Experiments using real images

A second series of experiments are conducted using real images. The three methods are used on the same first real image, and the results are shown in Figures 5-7. The radial distortion parameters extracted are listed in Table 1. As can be seen from Table 1 and Figures 5-7, significantly different sets of values of radial distortion parameters result in similar-looking undistorted images. Figure 8

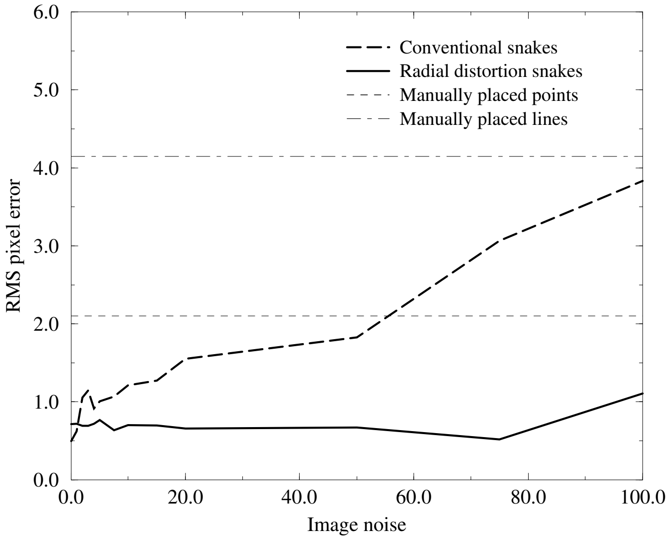


Figure 4: Graph showing the effect of gaussian image noise (standard deviation in intensity level) on RMS undistortion error E_{RMS}.

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.

Another example involving a real image is shown in Figure 9. This scene is a more difficult one due to the relative sparseness of structurally straight and long lines within the camera viewing space. As a result, the recovery of the radial distortion parameters are more sensitive to errors in the extracted curves. This is evidenced by Figure 9(d) for the conventional snake algorithm, where slight errors have resulted in significantly erroneous estimates in the radial distortion parameters.

Method	κ_1	κ_2
Manual	2.93×10^{-7}	7.10×10^{-12}
Conventional snakes	-5.30×10^{-7}	1.58×10^{-11}
Radial distortion snakes	5.47×10^{-7}	3.55×10^{-12}

Table 1: Comparison of radial distortion parameters from the three calibration methods for the office scene image.

(a) (b) (c)

Figure 5: Office scene (manual): (a) Original distorted image, (b) Selected points, (c) Undistorted image.

(a) (b) (c)

Figure 6: Office scene (using conventional snakes): (a) Chosen snakes, (b) Final snake configuration, (c) Undistorted image. Note the relatively uneven snake to the left of image in (b).

(a) (b) (c)

Figure 7: Office scene (using radial distortion snakes): (a) Chosen snakes, (b) Final snake configuration, (c) Undistorted image. Notice the smoother snakes compared to Figure 6.

(a) (b) (c)

Figure 8: Office scene (comparing the two snake implementation): (a) Initial snake configuration, and final snake configuration for (b) Conventional snakes, (c) Radial distortion snakes.

On the other hand, the radial distortion snake algorithm resulted in a visually more correct undistorted image.

Radial distortion snakes appear to have the effect of widening the range of convergence compared to conventional snakes (as exemplified by Figure 8). It is reasonable to hypothesize that radial distortion snakes have fewer false local minima, although this number will increase with the number of radial distortion parameters to be estimated. An example of radial distortion snakes converging to a wrong local minima is shown in Figure 10. As can be seen, due to the rather bad placing of the initial snakes (specifically the two most vertical ones), the snakes in both implementations converged to straddle different parallel edges, causing incorrect estimated radial distortion parameters.

4 Discussion

The first method of manually picking discrete direct line points is the simpliest to implement and understand, but of all the three methods described, it is the most burdensome to the user. In our implementation, the user has to be relatively careful in choosing the points. This is because the user-designated input locations are used directly in the radial distortion parameter estimation step. One can, however, add the intermediate process of automatic local edge searching and location refinement, but this may pose a problem in an image of a complicated scene with many local edges.

For the two methods that uses snakes, it is clear from experiments that using the radial distortion snakes is better than using conventional snakes. We have demonstrated that the radial

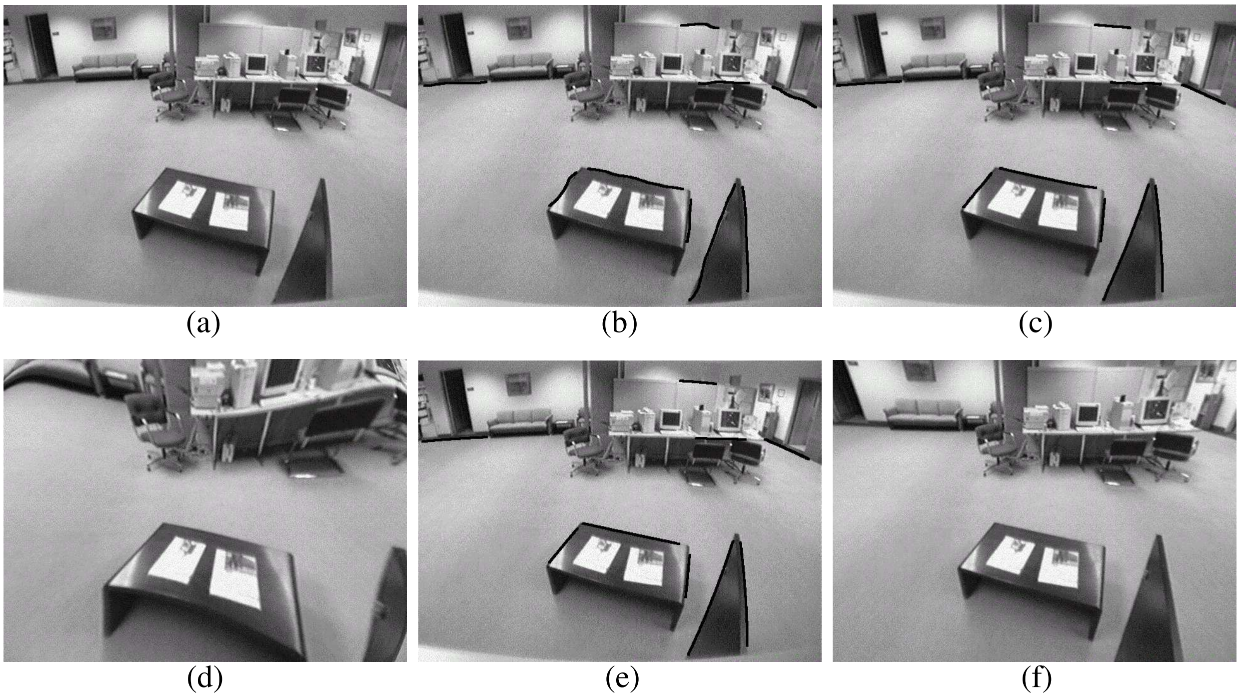


Figure 9: Another example: (a) Original image, (b) Initial snake configuration, and final snake configuration for (c) Conventional snakes, (e) Radial distortion snakes. The respective undistorted images are (d) and (f). Note that the snakes are shown in black here.

distortion snakes find best adaptation according to best global fit to radial distortion parameters. They appear to have fewer false local minima in comparison to conventional snakes, and are less prone to being trapped in bad local minima. 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* [9].

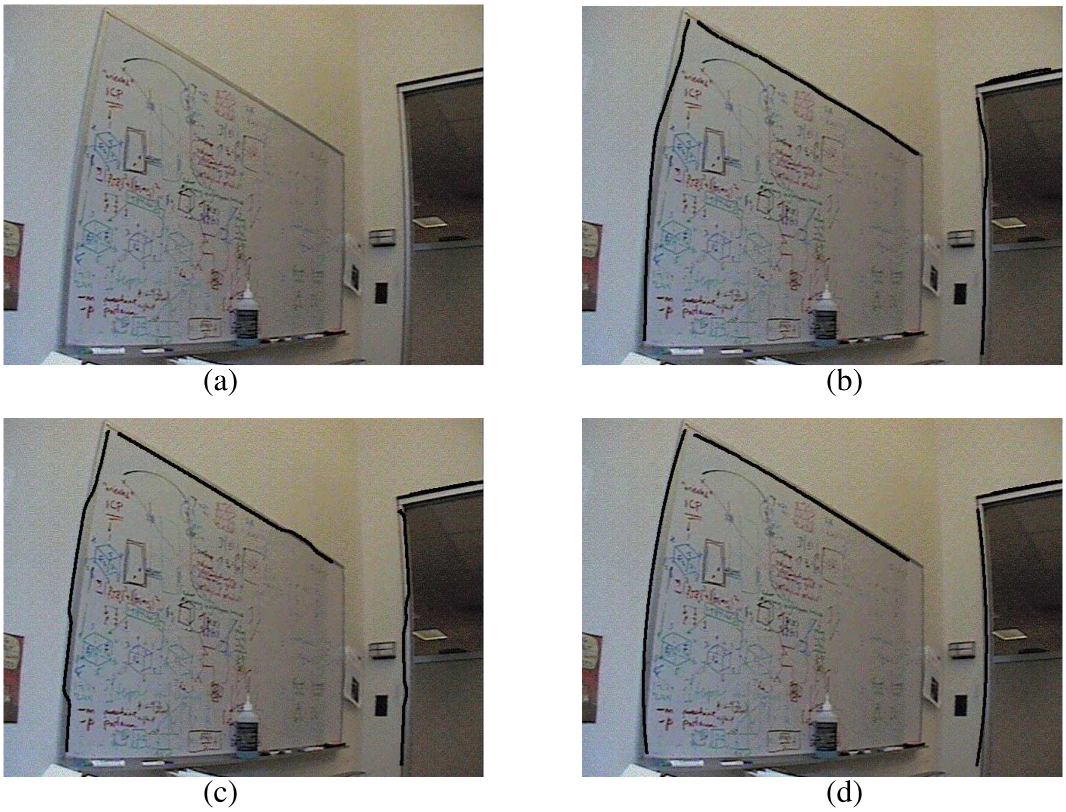


Figure 10: Snake failure example: (a) Original image, (b) Initial snake configuration, and final snake configuration for (c) Conventional snakes, (d) Radial distortion snakes. Note that the snakes are shown in black here.

5 Summary and future work

We have described three semiauthomatic methods for estimating radial distortion parameters from a single image. All the methods rely on the user indicating the edges of projected straight lines in space on the image. The first requires good user point to point localization while the other two only requires approximate initial snake configurations.

In essence, all the methods use the point locations to estimate the radial distortion parameters based on least-squares minimization technique. However, the last two methods automatically deforms the snakes to adapt and align along to strong edges subject to regularization. In the second method, the snakes behave independently and in a conventional way with internal smoothing constraints and external edge-seeking forces. In the third method, the snakes (called radial distortion snakes) behave coherently to external edge-seeking forces and more importantly, directly linked to the image radial distortion model. As a result, this method is more robust and accurate than the other two methods.

One direction for future work is to extend this work to estimate the principal point and tangential (or decentering) distortion parameters in addition to the radial distortion parameters. This would come at the expense of higher complexity and potential instability or problems in convergence. 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 [7]).

References

[1] S. Becker and V. B. Bove. Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In *Proc. SPIE Visual Data Exploration and Analysis II*, volume 2410, pages 447–461, February 1995.

[2] D. C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 32(3):444–462, May 1966.

[3] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, August 1971.

[4] I. Carlbom, D. Terzopoulos, and K. M. Harris. Reconstructing and visualizing models of neuronal dendrites. In N. M. Patrikalakis, editor, *Scientific Visualization of Physical Phenomena*, pages 623–638. Springer-Verlag, New York, 1991.

[5] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, November 1993.

[6] W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41(12):1479–1486, December 1975.

[7] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[8] S. R. Gunn and M. S. Nixon. A robust snake implementation; A dual active contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):63–68, January 1997.

[9] K. Ikeuchi and M. Hebert. Task oriented vision. In *Image Understanding Workshop*, pages 497–507, Pittsburgh, Pennsylvania, September 1990. Morgan Kaufmann Publishers.

[10] S. B. Kang and R. Szeliski. 3-D scene data recovery using omnidirectional multibaseline stereo. In *Proc.s IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 364–370, June 1996.

[11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision (ICCV'87)*, pages 259–268, London, England, June 1987. IEEE Computer Society Press.

[12] D. L. Light. The new camera calibration system at the U.S. Geological Survey. *Photogrammetric Engineering and Remote Sensing*, 58(2):185–188, February 1992.

[13] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH'95)*, pages 39–46, August 1995.

[14] Chester C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, Virginia, fourth edition, 1980.

[15] G. P. Stein. Lens distortion calibration using point correspondences. A. I. Memo 1595, Massachusetts Institute of Technology, November 1996.

[16] R. Szeliski. Image mosaicing for tele-reality applications. In *IEEE Workshop on Applications of Computer Vision (WACV'94)*, pages 44–53, Sarasota, Florida, December 1994. IEEE Computer Society.

[17] R. Szeliski and S. B. Kang. Direct methods for visual scene reconstruction. In *IEEE Workshop on Representations of Visual Scenes*, pages 26–33, Cambridge, Massachusetts, June 1995.

[18] C. J. Taylor, P. E. Debevec, and J. Malik. Reconstructing polyhedral models of architectural scenes from photographs. In *Fourth European Conference on Computer Vision (ECCV'96)*, volume 2, pages 659–668, Cambridge, England, April 1996. Springer-Verlag.

[19] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models and 3D object reconstruction. *International Journal of Computer Vision*, 1(3):211–221, October 1987.

[20] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, August 1987.

[21] G.-Q. Wei and S. D. Ma. Implicit and explicit camera calibration: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):469–480, May 1994.

[22] A. L. Yuille, D. S. Cohen, and P. W. Hallinan. Feature extraction from faces using deformable templates. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'89)*, pages 104–109, San Diego, California, June 1989. IEEE Computer Society Press.

digital™

Semiautomatic Methods for Recovering Radial Distortion Parameters from A Single Image

Sing Bing Kang

CRL 97/3

May 1997