Automated Calibration and Correction of Radial Lens Distortion

by

Blair Prescott
B.A.Sc., University of British Columbia, 1992

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF APPLIED SCIENCE
in the
Department of Mechanical Engineering

We accept this thesis as conforming
to the required standard

[redacted]

Dr. G. McLean, Supervisor (Dept. of Mechanical Engineering)

[redacted]

Dr. C. Bradley, Departmental Member (Dept. of Mechanical Engineering)

[redacted]

Dr. F. El-Gujbaly, Outside Member (Dept. of Electrical Engineering)

[redacted]

Dr. M. Van Emden, External Examiner (Dept. of Computer Science)

University of Victoria

Supervisor: Dr. G. McLean

Abstract

Lens distortion is present in all cameras, especially when inexpensive lenses are used. Distortion can significantly affect geometrical image properties, degrading the performance of applications which use geometric measures, such as shape recognition, object tracking, robot guidance, and videometry. This thesis describes automated techniques to measure and correct for distortion using image feature extraction, optimization, and image warping.

Examiners:

[redacted]

Dr. G. McLean, Supervisor (Dept. of Mechanical Engineering)

[redacted]

Dr. C. Bradley, Departmental Member (Dept. of Mechanical Engineering)

[redacted]

Dr. F. El-Guibaly, Outside Member (Dept. of Electrical Engineering)

[redacted]

Dr. M. Van Emden, External Examiner (Dept. of Computer Science)

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank my supervisor, Dr. Ged McLean, for his invaluable assistance in the preparation of this thesis. Thanks also to Alan Walford of Eos Systems Inc., Dayle Kotturi, Dave Gawley, and Dan Pifko for encouragement and technical assistance.

Chapter 1

Introduction

1.1 Videometry

Videometry and photogrammetry refer to the use of multiply redundant images of an object to derive an accurate, three-dimensional representation of the object's principal dimensions. Videometry is an emerging technique which utilizes digital imaging technology, image processing, and computer vision to automate traditional methods of film-based photogrammetry. Objects which are difficult to measure by any other means due to size, inaccessibility, fragility, or complexity of shape, can be accurately measured based on one or more video images taken of them [34]. Videometric techniques are used in many applications from a wide variety of disciplines, including measurement of the surface of the earth and other celestial objects from aerial and satellite images [14, 17, 18], control of assembly line automation [35], and sensing the surroundings of autonomous vehicles [11].

The principal advantage videometry holds over photogrammetry is the opportunity to use powerful digital computers to replace the more time-consuming and in-

accurate manual measurement of large-format photographs. Using computers, more sophisticated algorithms can be used to locate features of interest in images, to match these features among several images, and to compute the position in space of the object in the scene which corresponds to the features.

A typical videometric process usually includes the following steps:

Imaging A number of pictures of the target are taken from different viewpoints and stored in a digital format for later processing. The imaging process takes information about a three-dimensional scene, and encodes it as a two-dimensional image, so some information is missing from each image.

Feature Extraction is carried out by a computer to accurately locate important, recognizable features in each of the images.

Feature Matching Manual or automated procedures are used to equate identical features which occur in multiple images. This provides multiple observations of each scene location.

Reconstruction Based on the exact position of the common features in each of the images, the positions in space of each of the features can be calculated by back-projecting from the image planes into the scene and locating common intersections.

Before the reconstruction of feature points in the image can take place, however, a model of the imaging transform, which projects a three-dimensional space onto a two-dimensional space, must be known. The process of discovering the parameters of this model, camera calibration, is critical to the accuracy with which the image features can be located.

1.2 Camera Calibration

Image formation can be modelled as a transformation between points in the three-dimensional 'world frame' and points in the two-dimensional 'image frame.' In an ideal pinhole camera, each image point displays the world point which lies on a line connecting the image point with the pinhole, as shown in Figure 1.1.

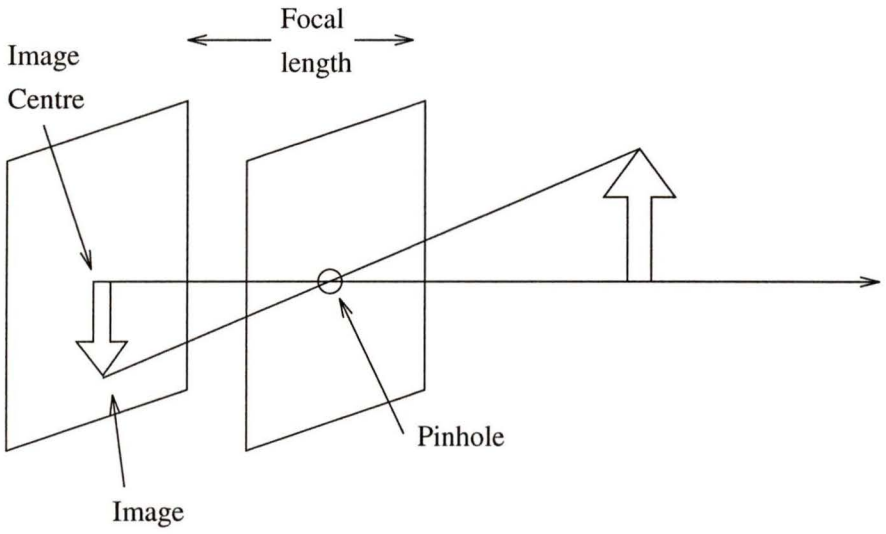

Figure 1.1: Pinhole camera.

Given multiple views of the same feature point from different camera positions, it is possible to use the intersection of these lines from the two-dimensional images to locate the position of the feature in the three-dimensional world frame, as shown in Figure 1.2. This is one way in which our two eyes allow us to determine depth information.

Underlying this technique, however, is the fact that certain parameters of the cameras must be accurately known or 'calibrated' before any information about the scene can be determined. For pinhole cameras, the needed parameters are:

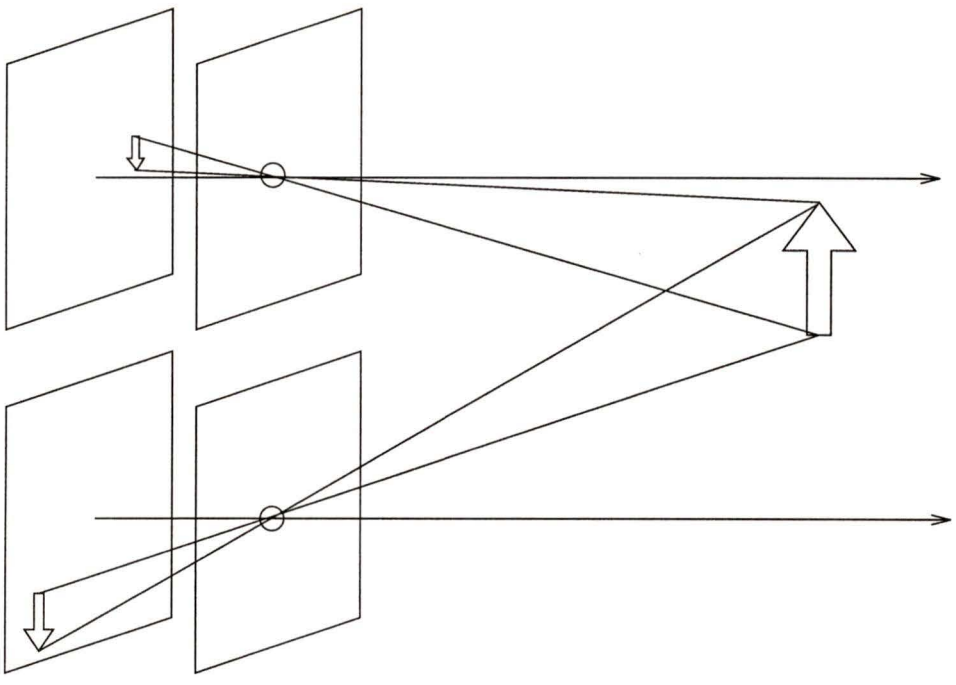

Figure 1.2: Stereo pinhole cameras.

Focal Lengths The distances between the image planes and the pinholes.

Camera Positions The distance and direction from each camera position to some reference point.

Camera Orientations The direction in which the cameras' optical axes are aimed.

Image Centres The points on the image planes intersected by the optical axes.

These parameters must be known to allow the location of scene features. Without knowing the relationship between the images, it is impossible to merge the information contained in them. As well, the accuracy of measurement is limited by the accuracy with which the cameras can be calibrated. A number of techniques for camera calibration will be reviewed in Chapter 2.

1.3 Distortion

Distortion, which is another consideration in both videometry and photogrammetry, is caused by different amounts of magnification at different points in the image. Distortion is present to some degree in all cameras, but more so in the inexpensive video cameras that are widely used today. Some examples of a common type of distortion, radial lens distortion, are shown in Figure 1.3. Depending on whether the magnification increases or decreases toward the edge of the image, it is also known as 'barrel' or 'pincushion' distortion, after the curved lines which it results in. Measurements made on images suffering from this curvature may have reduced accuracy, due to the deviation from the pinhole camera model.

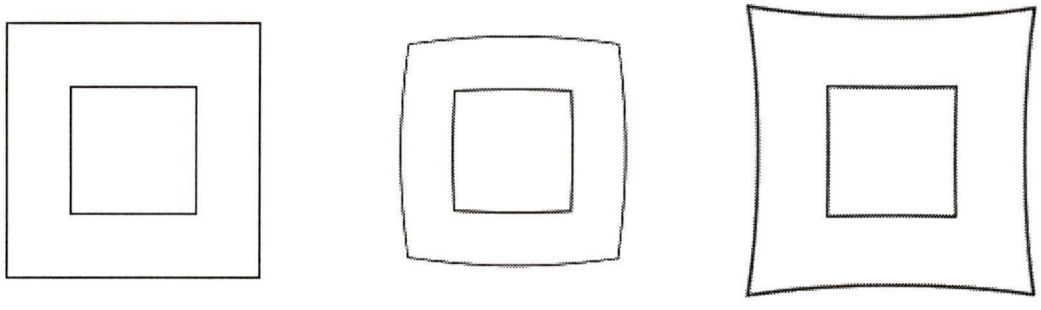

Figure 1.3: Types of radial distortion. Left: Original image. Centre: Barrel distortion. Right: Pincushion distortion.

Every application which makes geometric measurements on an image, such as feature detection, relative position measurement, object recognition, or object tracking, should take into account the effects of distortion.

1.4 Overview of the Thesis

This thesis presents a novel method of accurately measuring the radial distortion in an image, and correcting the image in such a way as to remove the effects of the distortion. In this manner, it is possible to improve the accuracy of videometric measurements.

The method uses lines as image features, which can be detected in digital images to a high level of accuracy. Due to lens distortion, straight lines in a scene appear slightly curved. Optimization can be used to determine the degree of curvature of the lines in an image, making possible the correction of the distortion through image warping.

Chapter 2 will begin by describing camera calibration in more detail. Explanations of the various calibration parameters will be followed by a review of previous work in the calibration field. Chapter 3 will focus on lens distortion, providing descriptions of old and new methods of detecting and measuring distortion parameters. An outline of the proposed method will be found here.

In Chapter 4, algorithms for detecting linear image features, for measuring line curvature, and for correcting for distortion will be described. This chapter will conclude with a demonstration of the new algorithm on a synthetic image. Experimental results and performance evaluations will be given in Chapter 5, based on both real and synthetic image examples.

Finally, conclusions and ideas for future work will be presented in Chapter 6.

Chapter 2

Camera Calibration

The goal of camera calibration is to develop a mathematical model of the transformation between world points and image points, which is implicit in the image formation process. The world points are typically located in a three-dimensional world coordinate frame, while the image coordinates are measured in terms of pixel distances in the two-dimensional image. Since this is a many-to-one mapping, it is not possible to locate a single world point that corresponds to a given image location. Instead, the line connecting the image point and the centre of the lens can be projected, as shown in Figure 1.1. The world point corresponding to the image location is constrained to lie on this line.

The parameters which affect this mapping can be divided into three categories:

Extrinsic parameters describe the relationship between the camera frame and the world frame, including position and orientation information. These parameters are changed every time the camera is moved, and thus they may need to be measured frequently.

Intrinsic parameters describe the characteristics of the camera, and can be rel-

atively stable over time. They include such attributes as focal length, scale factors, and location of the image centre.

Distortion parameters describe the geometric non-linearities of the camera. Often, they are considered to be intrinsic parameters.

Each of these parameters is discussed in the following sections.

2.1 Extrinsic Parameters

Consider a pinhole camera arbitrarily positioned and oriented with respect to some world coordinate frame (X_w, Y_w, Z_w), as shown in Figure 2.1. The camera reference frame, centred at the aperture of the camera, is denoted (X_c, Y_c, Z_c), with the Z_c axis directed perpendicular to the plane of the aperture. The transformation between a point in the world frame and a point in the camera frame is given by [9]:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = R \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} + T$$

where R is a rotation matrix and T is the translation vector which separates the frames.

The R matrix can be completely specified by three parameters, (α, β, γ) such as the three Euler angles, or the roll, pitch and yaw angles of the camera frame . Likewise, the T vector adds three parameters (T_x, T_y, T_z) to the calibration variables. Together these six extrinsic parameters $(\alpha, \beta, \gamma, T_x, T_y, T_z)$ specify the camera's relationship to the world reference frame [36].

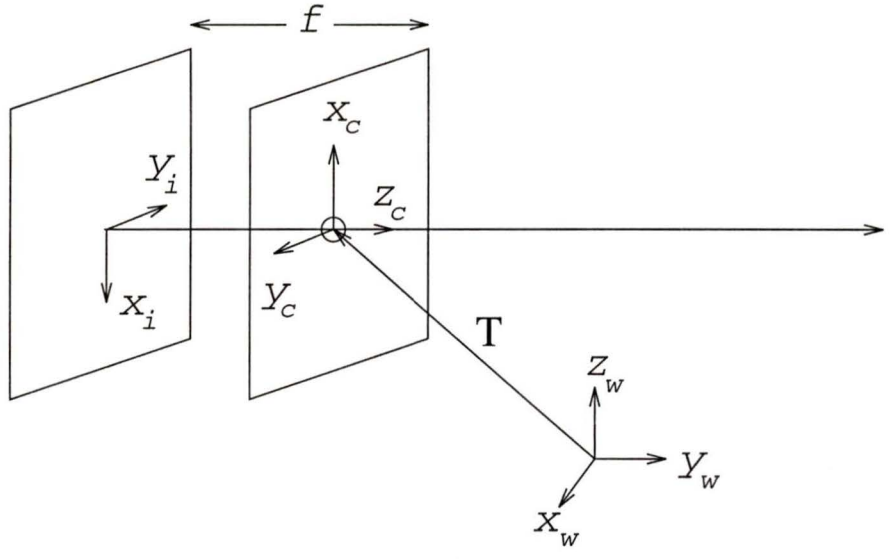

Figure 2.1: Coordinate systems.

2.2 Intrinsic Parameters

Next, consider the effects of the camera's structure on the images it generates. Each point (x_c, y_c, z_c) in the camera's field of view will be projected by the lens onto the image plane at a point (x_i, y_i) according to the perspective transform:

$$x_c = x_i \frac{z_c}{f} \qquad \text{and} \qquad y_c = y_i \frac{z_c}{f}$$

where f is the focal length, the distance from the lens to the image plane.

On the image plane, an array of sensor elements samples the image into a new coordinate system (X, Y), which corresponds to the pixel values that are recorded as the image.[1] At this step, a small sampling error is introduced, due to the finite pixel size. For the common case of a rectangular array of sensors, this transformation can

[1]There may also be a scaling of the image at the frame buffer, but the effects are indistinguishable.

be expressed by:

$$X = \frac{x_i}{d_x} + c_x \qquad \text{and} \qquad Y = \frac{y_i}{d_y} + c_y$$

where (d_x, d_y) give the horizontal and vertical pixel size, which are scale factors which convert from real-world dimensions to pixel dimensions and (c_x, c_y) give the position of the true image centre in pixel coordinates. Typically the true image centre is proximal to the centre of the sensor array, but due to manufacturing tolerances, it is always offset by a small amount, normally on the order of a few pixels [25]. This location will be important in the discussion of distortion parameters.

Together, the five parameters (f, d_x, d_y, c_x, c_y) constitute the intrinsic parameters associated with the camera.

2.3 Distortion Parameters

The final class of calibration parameters which affect image formation are related to the distortion properties of the lens assembly. Although modern lenses can be made with very high precision, typical low-cost lenses are found to exhibit significant distortion, especially in areas near the fringes of the image [40]. Distortion arises when image points are magnified by different amounts at different positions in the image. This distortion can be limited through appropriate lens design parameters, but it exists in all lenses to some extent.

The two principal forms of distortion considered in videometry and photogrammetry applications are radial and decentring distortion [6]. Radial distortion can be in the form of barrel or pincushion distortion, as described in Section 1.3. Decentring distortion is a result of misalignment of lens elements. Examples of radial and decentring distortions of a synthetic image are shown in Figure 2.2.

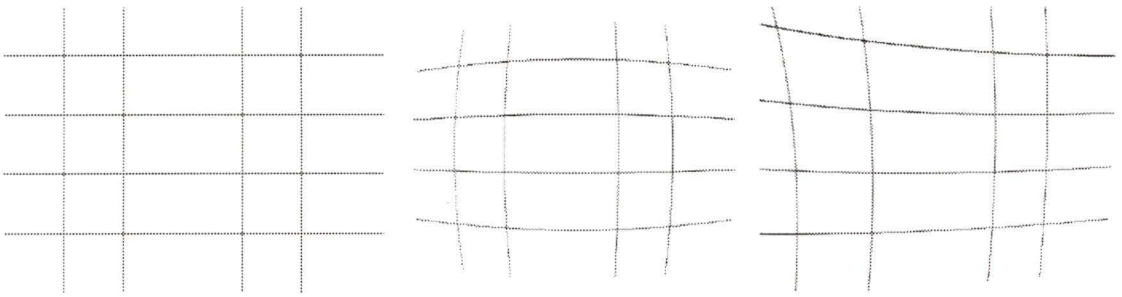

Figure 2.2: Distortion of a synthetic image. Left: Original. Centre: Radial distortion. Right: Decentring distortion.

The customary model for the correction of radial and decentring distortion [6] is given by the following equations. The coordinates (x, y) refer to a point in the original, uncorrected image, and (x', y') is the corresponding corrected location.

$$x' = x + \bar{x}(K_1 r^2 + K_2 r^4 + K_3 r^6 + \cdots) + [P_1(r^2 + 2\bar{x}^2) + 2P_2\bar{x}\bar{y}][1 + P_3 r^2 + \cdots]$$

$$y' = y + \bar{y}(K_1 r^2 + K_2 r^4 + K_3 r^6 + \cdots) + [P_2(r^2 + 2\bar{y}^2) + 2P_1\bar{x}\bar{y}][1 + P_3 r^2 + \cdots]$$

where

$$\bar{x} = x - c_x$$

$$\bar{y} = y - c_y$$

$$r = [(x - c_x)^2 + (y - c_y)^2]^{1/2}$$

and K_1, K_2, K_3 are the coefficients of radial distortion and P_1, P_2, P_3 are the coefficients of decentring distortion. r is the radius of an image point from the image centre, defined as (c_x, c_y) above.

Typically, only one or two distortion parameters are modelled, as the higher order terms are comparatively insignificant [40]. For the purpose of this investigation, the focus will be on the distortion parameters K_1 and K_2.

2.4 Previous Work in Camera Calibration

The present work traces its roots back to three distinct groups of previously published works. Most important is the work of Brown, a pioneer in the area of film photogrammetry, who first set out methods of correcting for image distortion. More recent influences are the vanishing point calibration methods, such as those described by Echigo and Wang, which use line-based methods for calibration of digital cameras, though without consideration of distortion. The largest of three groups is centred on the works of Tsai, who integrates calibration and simple distortion correction for camera calibration. The contributions of each of these groups are described in this section.

2.4.1 The Work of Brown

D. C. Brown, of DBA Systems in Melbourne, Florida, was one of the first published authors in the field of camera calibration. In a series of papers from 1956 to 1971, he described several methods of calibrating film cameras to an extremely high degree of accuracy (i.e. 1 part in 150,000 in all three axes of measurement [6, p. 23]) His methods use extremely good models of radial and decentring distortion, but require iteration and large matrix operations to reach a solution. Brown's methods include Stellar Calibration, Simultaneous Multiframe Analytical Calibration, Finite Target Calibration, Analytical Plumb Line Calibration, and Self Calibration.

Each of the methods mentioned above uses a large set of points measured from photographs. The typically lower resolution available in digital images would require some modifications to the methods. Also, while they appear to have excellent accuracy, computational complexity and the use of large calibration targets make these methods less appealing.

2.4.2 Vanishing Points

Two relatively recent papers use parallel lines and vanishing points to arrive at estimates of the position, orientation, and focal length of the lens. Neither of the methods considers distortion.

In [12], Echigo uses a prepared cubic target. An oblique view of any two sides provides three sets of at least five parallel lines. Aspect ratio and image centre are precalculated and assumed known. An iterative step is used to improve accuracy down to an estimated 0.2 pixels average calibration error in simulated images.

In [38], Wang uses a hexagonal target on the ground plane to establish three sets of two parallel lines. He also suggests using a pattern of square tiles to make up a hexagon, so a mobile robot could orient itself from many existing floor patterns. A Hough transform is used to locate lines. An accuracy of between one and five percent is claimed, with only quadratic equations to be solved and no iteration required.

The modest computational requirements, coupled with the simplicity of the targets required, make these methods attractive to implement. Provided good estimates of the intrinsic parameters can be obtained through other means, these algorithms could form the basis of a field calibration using building lines or other existing features.

2.4.3 Tsai and Others

With his 1987 paper [37], Tsai touched off a flurry of activity in the CCD camera calibration area. He provided an excellent literature review, then went on to describe several two-stage techniques that involve one or more images of a planar calibration target containing 60 points of known position. A limited model of radial distortion is used, and the horizontal scale factor can be recovered if multiple images from different object distances are used. Tsai's methods do involve solving systems of linear

equations, and include an iterative step that solves for two distortion coefficients, but they can be accomplished on a 68000-based MASSCOMP minicomputer in under 1.5 seconds. He claims accuracy on the order of 1 part in 4000 over the field of view and 1 part in 8000 in depth.

In a later paper [25], Lenz and Tsai gave methods for observing the horizontal scale factor and the image centre, as well as discussing line jitter. The scale factor is best determined by making direct measures of the camera pixel clock frequency and the A/D-converter sampling frequency, while the image centre can be determined directly using a laser aligned with the optical axis, or with a calibration target of known features and an iterative method.

Sid-Ahmed and Boraie [35] and Grosky and Tamburino [19] present two similar methods that claim to improve calibration accuracy and efficiency. They both use known target positions to correct for an improved distortion model by an iterative procedure.

In the most recent paper, Tsai's former co-author Lenz presents a simple calibration scheme consisting of four steps [24]. It requires pre-calibration of the horizontal scale factor and principal point, but requires only 15ms to process 36 coplanar calibration points. Only small improvements (3% average error, 30% maximum error) were obtained from subsequent nonlinear optimization. He also provides good discussions of line jitter, noise and systematic errors in calibration point location due to perspective and distortion.

2.5 Conclusion

This chapter has described camera calibration as being the determination of a set of intrinsic, extrinsic, and distortion parameters, and a number of prior algorithms

for determining these parameters were presented. In general, existing algorithms use large calibration targets, whose geometry must be accurately determined by some other means before camera calibration can take place.

The next chapter will focus on the distortion parameters, and present an outline of a new method for determining them accurately. Only simple, readily available calibration targets will be required, and no knowledge of the dimensions of the calibration target is needed.

Chapter 3

Calibration of Lens Distortion

3.1 The Importance of Lens Distortion

Any process which requires accurate measurements based on images requires consideration of the effects of lens distortion. As early as 1956, Brown was developing calibration methods to correct for lens distortion in photogrammetric cameras. A recent paper [39] points out that

> Determining the distorted image coordinates is important, especially when lens distortion is large. For example, the epipolar lines used in stereo matching should be distorted in accordance with the lenses' real distortion process to reduce the searching area.

Another author states [35, p. 512]

> ... if we add a supplementary positive lens to the lens system to allow a closer focussing range, or a polarizing filter to cut down on light reflection

(specular effects) from a metallic target, then higher order terms of radial lens distortion and tangential distortions have to be considered.

Unless an appropriate distortion model is used to correct for the effects of lens distortion, image points will be offset from their expected positions on the image plane. For inexpensive cameras, this can change the dimensions of the image by as much as 5% over the field of view, making more accurate measurements impossible.

3.2 Methods for Measuring Distortion

Many different approaches have been used to determine the appropriate distortion parameters associated with a camera system, but they all tend to use the same property of lens distortion. This property is that lens distortion changes the distance between image points. Thus, existing methods all focus on location of points in calibration images. In contrast, this thesis describes a method based on the property that straight lines in the scene appear curved in the distorted image. The important differences between these 'point-based' methods and the new 'line-based' method are described in this chapter.

3.2.1 Point Correspondence Methods

In a typical point-based method, a calibration target consisting of a set of regularly spaced, identical features is imaged by a camera. The image locations of well defined points, such as corners [36, 40] or centroids [13] are then located to sub-pixel accuracy. Using the known world locations of the points and the corresponding image locations, an optimization process is used to determine all the internal, external, and distortion parameters defined by the model in use.

Various authors provide improvements to this general model, which allow some parameters to be solved for explicitly, reducing the number of variables requiring optimization [36]. This improves the rate of convergence to a solution in the optimization process.

A principal disadvantage to this type of method is the requirement for a precisely constructed calibration target of known dimensions. These targets can be expensive and time-consuming to produce, ill-suited for on-site calibration, and may be sensitive to changes in temperature.

One way to circumvent this problem is to use a set of targets whose positions are unknown, and to solve for the positions of the targets as part of the optimization process [5]. However, this leads to a large number of variables to be optimized, and requires more redundant images of the scene be analyzed, as well as more sophisticated algorithms for optimization to ensure convergence in a reasonable time.

The line-based method described in the next section overcomes some of these problems. It does not require measured targets, a large number of images, or optimization of a large number of variables.

3.2.2 Line Straightness Method

The perspective projection of a straight line in a three dimensional scene domain is itself a straight line in the two dimensional image domain [5][1]. This means that curvature of lines in the image that should be straight is due only to lens distortion.

Determining the distortion model coefficients can be visualized as laying a straight line over the 'warped' image points, and then adjusting the distortion of the straight

[1]Except for the uninteresting collinear case where the projection is a point.

line until it matches the image line. Optimization can be used to find the distortion model parameters which minimize the deviation between the image data and the synthetically warped theoretical line.

The requirements for calibration images, therefore, are greatly reduced. Instead of requiring precise measures of scene distances, all that is required is the presence of straight lines. Even parallelism is unnecessary. A set of vertical blinds, the edge of a building, or a simple ruler all make good targets. It is not required that the position and orientation of the lines be known beforehand, as these are additional variables determined by the optimization process.

Another advantage of this process is that since no fixed spatial relationship between lines in the scene is required, lines from a series of images can be combined to form a composite image. This allows data from several images of a single line to be analyzed as though it came from a single image of many lines. This is useful in building up the sets of lines required to guarantee convergence of the optimization algorithm.

One disadvantage to this process is that it does not, by itself, allow the determination of the full camera calibration. Some applications, such as art restoration, may not require the added complexity of complete calibration. For those that do, it is worth observing that by decomposing the distortion from the other calibration parameters, the line-based method allows a simple, distortion-free camera calibration method to be used accurately. For example, [12] provides a method based on lines and vanishing points, which could take advantage of the accurate line equations that are found as a side-effect of the line-based distortion measurement.

3.3 Automated Distortion Correction

This section will attempt to give a brief outline of the proposed technique for the correction of lens distortion. Figure 3.1 shows the principal operations involved in the calibration of a camera and the correction of images, which are described in turn below. Details of the operations involved will be given in Chapter 4.

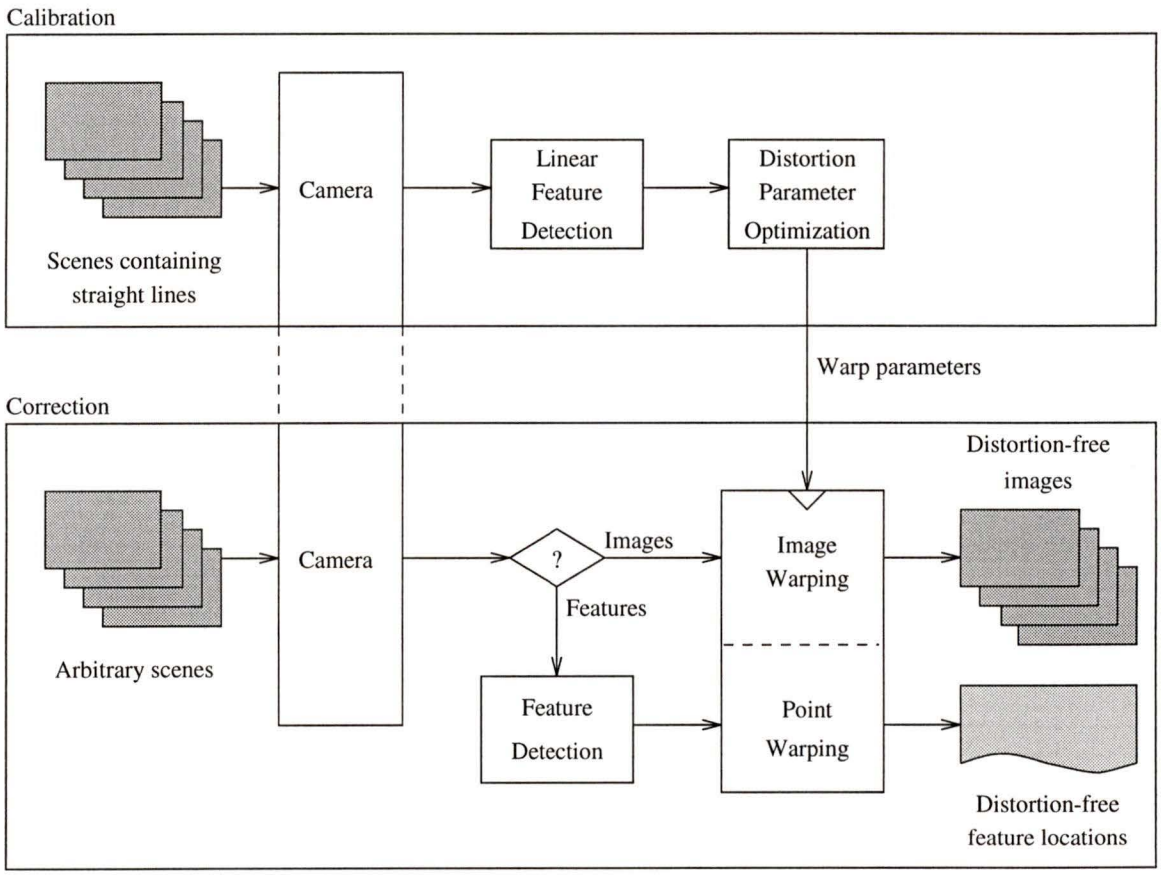

Figure 3.1: Automated distortion correction process.

3.3.1 Calibration Phase

In the calibration phase, a set of images are taken of a scene or scenes. The only requirement on the content of the scenes is that they must contain a number of obvious linear edge features, which are distributed over the scene with a variety of positions and orientation. Due to lens distortion, the images of these line features will appear curved in the resulting images.

The calibration images are passed to a process which extracts the linear features from the scene. This line detection process can be summarized in saying that adjacent pixels having similar gradient orientation are grouped together into Line Support Regions, or LSRs. These LSRs are described by lists of pixels which are members of each region.

Following the extraction of the LSRs, the lists of pixels are used as input to an optimization process. This iterative algorithm attempts to find the set of distortion parameters which most accurately describes the curvature of the lines in the original images. Once these parameters have been determined, the calibration process is complete.

3.3.2 Correction Phase

In the correction phase, completely arbitrary images can be taken by the calibrated camera. There are no restrictions on the content, only that the images be taken by the same camera, with the same lens settings. These images suffer the same distortion effects as the calibration images taken previously.

Using the distortion parameters found in the calibration phase, geometric image warping is used to correct for the distortion introduced by the camera. The resulting

images closely approximate the ideal image that would result if lens distortion were not present.

Alternately, the warping function can be applied only to the locations of points of interest, following some form of feature extraction. This is a more computationally efficient process than correcting the entire image and then performing feature extraction, since only a few points need be corrected, instead of every point in the image.

3.4 Conclusion

Three issues have been explored in this chapter. First, this chapter has attempted to demonstrate the need to correct for lens distortion, in order to improve the accuracy of procedures which make geometric measurements on images. Following this, a description of existing point-based techniques and the proposed line-based technique for the calibration of lens distortion was given. Fundamental differences between the two techniques were summarized. The chapter closed with a schematic description of the propsed technique, which described the tools needed for successful implementation.

The next chapter will delve deeper into the three fundamental tools required for the proposed technique: detection of linear features, optimization of distortion parameters, and geometric image warping. It will also present the first examples of the technique in use on synthetic images.

Chapter 4

Toolkit

The automated distortion correction method described in Chapter 3 requires completion of three major tasks for its operation:

- Detection of linear features in images.
- Optimization of the warping function to correct for the distortion present in the linear features.
- Warping of images to eliminate distortion.

This chapter will describe the algorithms used to accomplish each of these tasks, and present some examples of their use on synthetic images.

4.1 Detecting Linear Features

The line finding strategy used is based upon the method proposed by Burns [7] and employed in other more recent work which directly estimates line equations in image

analysis [27]. Individual pixels are grouped as belonging to a particular Line Support Region (LSR) based on the joint criteria of similarity of local gradient orientation and spatial connectedness. These LSRs are detected using a connected components analysis on an image of quantized gradient orientations. They form the basis for line equation estimation using a principal component analysis of the LSR data.

The quantization of gradient orientations into ranges of similar orientation is problematic given that there is no a priori knowledge of the relative distribution of lines in the image: well formed LSRs can be split between two levels of orientation quantization. Burns addressed this problem by forming two independent quantizations of gradient orientation with overlapping quantization levels, followed by two independent connected component operations. The final LSR's were formed by a voting process which decided which of the two redundant segmentations would be selected for any particular pixel. For the present work, it was noted that images containing strong line structure typically have a histogram of gradient orientation which is highly peaked, indicating the orientations of the dominant lines. N-level quantization algorithms have been proposed for some forms of simple image compression [4, 33, 32]. These create a set of quantization thresholds through the analysis of an image histogram and are used here to quantize the gradient orientations into N levels associated with strong lines.

The gradient magnitude (∇_{ij}) and orientation (Φ_{ij}) at each pixel are computed using a simple 2x2 gradient convolution mask. The histogram of weighted gradient orientations is computed by selecting an angular step size θ and choosing a set of k bins, where $k = \frac{2\pi}{\theta}$. The histogram bins the distribution of orientations:

$$H(n) \;=\; \sum \nabla_{ij}$$

for all i, j such that $((n-1)\theta - \pi) \;<\; \Phi_{ij} \;<\; (n\theta - \pi)$ and $0 < n < k$.

A distribution which describes the curvature of the histogram is formed by subtracting a local average of the histogram from the initial histogram function. The local average is computed cyclically to avoid edge effects at orientation extremes:

$$\hat{H}(n) = \frac{mH(n)}{\sum_{i=\frac{-m}{2}}^{\frac{m}{2}} H(n+i)} \tag{4.1}$$

Negative going zero crossings of the curvature function are detected to produce a set of thresholds Q_j, $0 < j \leq P$. These P thresholds are then used to quantize the image orientations so that the LSRs can be detected with connected components analysis [21]. The selection of orientation quantization thresholds using histogram curvature analysis allows for the number and orientation of quantization levels to be determined *by the image content* rather than by some arbitrary expectation. In addition, using the histogram to obtain quantization levels results in a simpler line finding strategy: the quantization and connected components analysis are only computed once for the image (compared to twice in previous line finding schemes) and there is no requirement for any voting process to determine correct LSR membership for individual pixels [28].

The combination of local gradient magnitude and orientation computation, N-level histogram quantization and connected components analysis produces a segmented image consisting of a set of M LSRs. Each region is described by a set of K_m pixels:

$$LSR_m = \{(x_k, y_k), k = 1, \ldots, K_m\} \tag{4.2}$$

The overall method of obtaining LSRs from image data works without supervision and requires only a single pass through the image for quantization followed by a single connected components analysis. An example of using this method for fitting lines to LSRs for sub-pixel accuracy is given in [29]. However, for this thesis, we make use of the LSRs themselves. Overall, the scheme is quite efficient and robust for marking well-defined image lines.

4.2 Optimization of the Warp Parameters

The last section described the method used to detect LSRs, which describe distorted line regions by sets of spatially connected pixels. Given those sets, this section will present a method for determining the corresponding warp parameters to correct the distortion of the line regions, as well as discovering the resulting line equations.

First, however, a word about the convention used for line equations. In describing the position and orientation of arbitrary lines in an image, the familiar slope and intercept description, $y = mx + b$, leaves much to be desired. Vertical lines are impossible to represent, and singularities occur in calculations involving near-vertical lines. Instead, the parametric formulation is used, where every point (x', y') on a given line satisfies the equation [1]:

$$x' \cos\theta + y' \sin\theta = \rho$$

where ρ is the perpendicular distance from the origin to the line and θ is the angle this line makes with the horizontal axis, as shown in Figure 4.1.

4.2.1 Problem Specification

Given a set of M LSRs, each described by a set of K_m pixels:

$$LSR_m = \{(x_k, y_k), k = 1, \ldots, K_m\} \tag{4.3}$$

and a warping function, which maps (x, y) to (x', y'), determine the set of warp parameters which will straighten out the curvature of the line, thus minimizing the total deviation from the line equations

$$x' \cos\theta_m + y' \sin\theta_m \approx \rho_m$$

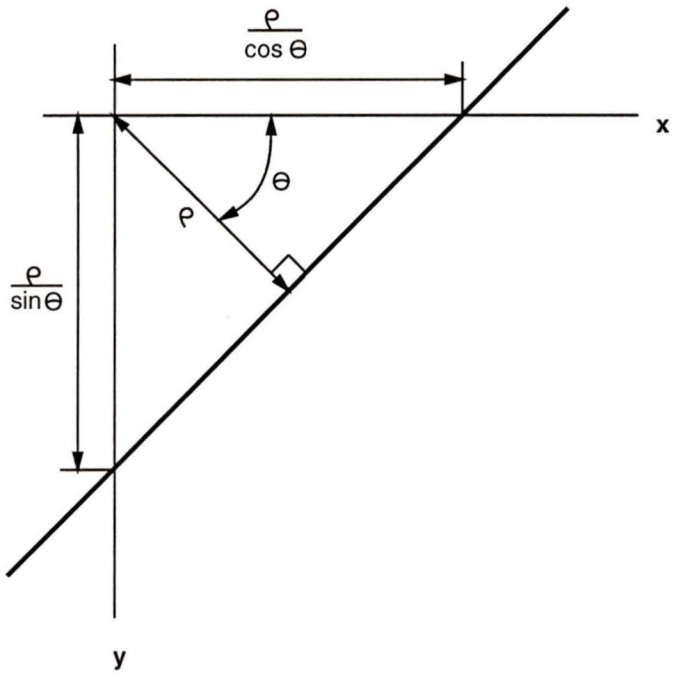

Figure 4.1: Line representation conventions.

4.2.2 Design Variables

In order to reduce the project to a manageable scope, only the warp parameters K_1, K_2, K_3, x_p, y_p are considered in the optimization. These parameters describe radial distortion and the image centre location, as mentioned in Chapter 2. In practice, these account for the majority of the distortion in modern lenses, and the other terms represent relatively small deviations. This results in warp equations of the form:

$$x' = x + \bar{x}(K_1 r^2 + K_2 r^4 + K_3 r^6)$$

$$y' = y + \bar{y}(K_1 r^2 + K_2 r^4 + K_3 r^6)$$

where

$$\bar{x} = x - x_p$$

$$\bar{y} = y - y_p$$

$$r = [(x - x_p)^2 + (y - y_p)^2]^{1/2}$$

We also assume that the undistorted position and orientation of each line is unknown, and thus must be discovered through optimization. Thus, the five warp variables, plus ρ_m and θ_m for each line in the image make up the set of design variables.

The initial estimates of the parameters are given by $K_1, K_2, K_3 = 0.0$, and (x_p, y_p) at the centre of the image. (ρ_m, θ_m) are initially estimated by least squares linear regression.

4.2.3 Objective Function

For the objective function, the cost associated with a particular choice of the design variables can be calculated by warping the image according to the chosen parameters, and summing the squared distance from each point to the corresponding line:

$$F = \sum_{m=1}^{M} \sum_{k=1}^{K_m} (x'_k \cos\theta_m + y'_k \sin\theta_m - \rho_m)^2$$

where the values for (x'_k, y'_k) are calculated from the warp equations above.

4.2.4 Constraints

Constraints on the design variables were also considered, since the ρ_m and θ_m values should result in line equations that pass through the image and the x_p and y_p values should be near to the middle of the image. However, it was found that enforced constraints added unnecessary complexity to the problem. Convergence is faster and more reliable if the constraints are not checked on each iteration of the optimization routine.

4.3 Image Warping for Distortion Correction

Image warping, or geometric transformation, is used in many applications to reshape images without significantly changing their colour or intensity characteristics. In the most common use, images taken with different sensors, at different times, or from different sensor positions or orientations are transformed so that they can be accurately overlaid with one another. This process allows cartographers to make maps from multiple satellite images [17], and surgeons to put together X-ray and magnetic resonance images to see a more complete diagnostic picture of their patients [23, 30]. In the computer graphics area, the geometric mapping of surface textures onto objects gives them a lifelike appearance without modelling the microscopic detail of the surfaces [3, 22]. Finally, the 'morphing' process that has been adopted by the television and motion picture special effects industry is also an example of a geometric transform [2, 20, 31].

In the context of lens distortion, image warping is also a useful tool. In this section, the process of correcting lens distortion will be formulated as a geometric transform between a distorted image and an undistorted image. Following a general discussion of geometric transforms and the mapping and interpolation issues involved, a discussion of the specific techniques used in correcting lens distortion will be presented.

4.3.1 Geometric Transformations

A geometric transformation of an image can be expressed as two functions:

$$x' = f(x, y)$$

$$y' = g(x, y)$$

where (x, y) are the coordinates of the original image and (x', y') are the coordinates of the transformed, or 'warped' image. The functions f and g define the mapping between the two coordinate systems. Simple rotation, scaling and translation of images, as well as more complex manipulations, can be accomplished with appropriate choices of mapping functions.

For example, a simple counter-clockwise rotation about the origin by an angle θ is given by:

$$x' = f(x, y) = x\cos\theta - y\sin\theta$$

$$y' = g(x, y) = x\sin\theta + y\cos\theta$$

This says that in order to rotate an image (x, y), scan through the image, and for each input pixel, write an output pixel with the same intensity value at the (x', y') position specified by the mapping function above. The resulting image will be a rotated version of the original.

Different choices of the parameter θ in the equations above result in different amounts of rotation of the image. More complex warping functions may have any number of parameters, each of which must be determined before the warp can be used.

The above mapping function is *global*, in that it is defined over the entire image. Some applications use an assortment of piecewise functions, such as splines, each of which is defined over a small region of the image. This allows added precision in cases when the desired geometric transform changes over the image, such as in the case of image registration based on many control points [15, 16, 17]. The disadvantage to this type of approach is that many parameters are required to specify the nature of the transform.

4.3.2 Sampling and Interpolation Issues

Two other issues arise in the discussion of mapping functions. First, the values returned by $f(x, y)$ and $g(x, y)$ are not necessarily integers, so it is not obvious how to deposit the input pixel values in the output array to create the warped image. As well, some output pixels may never be visited, leaving holes in the output, or some pixels may be visited more than once. A variety of techniques are available to deal with these problems, such as inverse mapping, interpolation, and sampling.

Inverse Mapping

One way to circumvent the problem where output image pixels are visited zero or many times, is to scan through the output image and use the inverse of the above mapping functions to find the corresponding input pixels. For the example rotation given, solving for x and y results in the equations:

$$x = x' \cos\theta + y' \sin\theta$$

$$y = -x' \sin\theta + y' \cos\theta$$

By traversing the output array, and using these functions to locate the corresponding position in the input array, each output pixel is visited exactly once. The problem of non-integral pixel values still exists, but using a suitable interpolation scheme, an average of neighbouring input pixels can be used to find an approximate output value [41].

Unfortunately, for more complex mapping functions, this inversion is not always possible to perform explicitly, or it may have multiple solutions.

Interpolation

Interpolation is used to estimate an appropriate value between the known points in an array. In general, a weighting function which varies with distance from the desired location is used to scale the contributions of nearby values. Chapter 5 of [41] provides a number of algorithms that are useful for this purpose. For example, in bilinear interpolation, the values at the four points surrounding a desired location are weighted by the function:

$$f(x', y') = (1-a)(1-b) \cdot f(x, y) + a(1-b) \cdot f(x+1, y) + (1-a)b \cdot f(x, y+1) + ab \cdot f(x+1, y+1)$$

where $x + a = x'$ and $y + b = y'$, as shown in Figure 4.2.

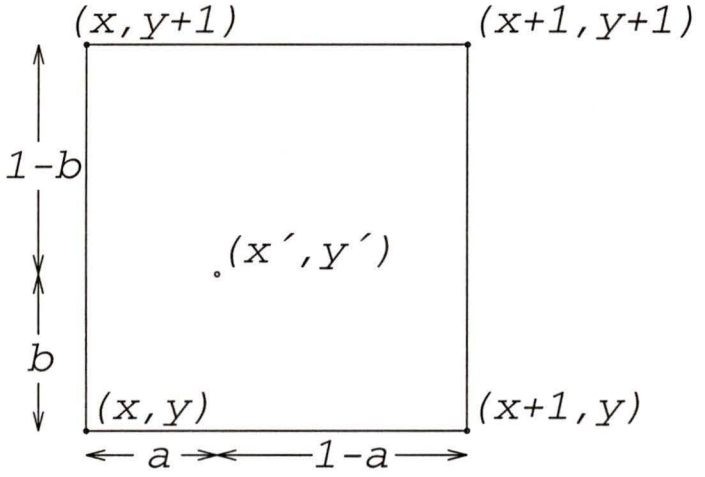

Figure 4.2: Bilinear interpolation.

Sampling Strategies

Another approach to avoiding holes in a forward mapping is supersampling [41]. With this technique, the input pixels are subdivided into smaller regions, and the centre of

each region is mapped to the output array. The number of subdivisions required to prevent holes depends on the degree of expansion of the input image, e.g. if the input image is being scaled by a factor of two in each dimension, dividing each pixel in half vertically and horizontally would be sufficient to prevent the occurrence of holes. For transformations where the degree of expansion changes throughout the image, there are also spatially varying sampling algorithms [26].

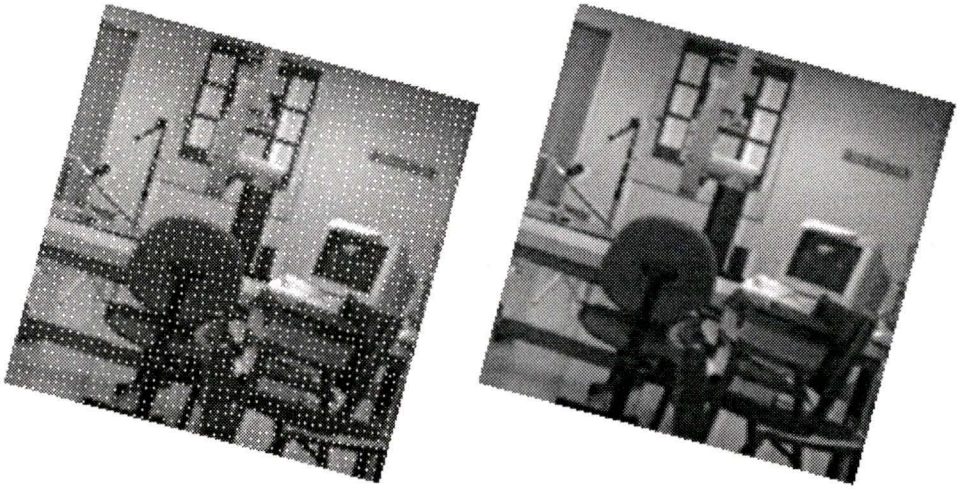

Figure 4.3: Supersampling to prevent holes.

Figure 4.3 shows an example of this technique. The image on the left has been rotated 15° using only simple nearest-neighbour interpolation. An array of holes appears as white speckles. The image on the right was generated using supersampling, and does not have any holes.

4.3.3 Image Warping for Lens Distortion Correction

In Chapter 2, a formulation for lens distortion was presented. If, for simplicity, the formula is restricted to the first two terms of radial lens distortion, the following is

the result:

$$x' = x + \bar{x}(K_1 r^2 + K_2 r^4)$$

$$y' = y + \bar{y}(K_1 r^2 + K_2 r^4)$$

where

$$\bar{x} = x - c_x$$

$$\bar{y} = y - c_y$$

$$r = [(x - c_x)^2 + (y - c_y)^2]^{1/2}$$

K_1 and K_2 are the coefficients of radial distortion and r is the radius of an image point from the image centre, defined as (c_x, c_y) above.

The form of these equations demonstrates that it is valid to think of the lens distortion as an image warping function. Thus, an image can be corrected for lens distortion provided that the values of the warp parameters K_1, K_2, c_x, c_y are known.

These equations describe a global, forward mapping from a distorted image, (x, y) to a corrected image (x', y'). In the next chapter, this will be seen to be very convenient for the determination of the warp parameters (K_1, K_2, c_x, c_y), but since it is a forward mapping, it proves to be inconvenient for the computation of the undistorted image based on those parameters. Unless special precautions are taken, holes will remain in the output image where output pixels have no corresponding input pixels.

To avoid the problems associated with forward mapping, it would be helpful if the inverse of the above functions was easily found. Unfortunately, it would only be possible to compute a numerical solution to the equations, which would be time consuming.

Instead, supersampling is used to avoid holes in the output image. For reasonable amounts of lens distortion, the expansion factor is always less than two, so dividing

each pixel into 4 smaller square regions prevents the formation of holes. The only disadvantage is that extra computation time is required, due to calculating the mapping function four times per input pixel instead of only once.

One other notable point about the equations given above is that they also define an inverse mapping, but from a distorted image to an undistorted image. This may seem like a useless result, but it does provide some simplification when adding synthetic lens distortion to an image for testing.

4.4 Overview of the Method

So far in this chapter, three important parts of the Automated Distortion Correction algorithm shown in Figure 3.1 have been described: linear feature detection, distortion parameter optimization, and image warping. Now, an example of the use of these tools on an image will be presented.

Consider the following example images. The first, Figure 4.4, shows a synthetic scene of evenly spaced, single-pixel width straight lines, each with a different intensity value. Taking an image of this 'scene' with a camera which suffers from lens distortion with parameters $K_1 = 2 \times 10^{-6}$, $K_2 = 3 \times 10^{-12}$, and with the image centre near the middle of the image at $(330, 240)$ results in the image shown in Figure 4.5.

The process of line detection is quite simple for this image, and basically amounts to reading off the coordinates of each pixel of similar colour. The resulting LSRs are then used in the optimization process to approximate the warping parameters needed to correct the warped image.

For this example, the optimization process results in the parameters $K_1 = 2.0081 \times 10^{-6}$, $K_2 = 2.8648 \times 10^{-12}$, and the image centre at the point $(331.7, 239.44)$, which are

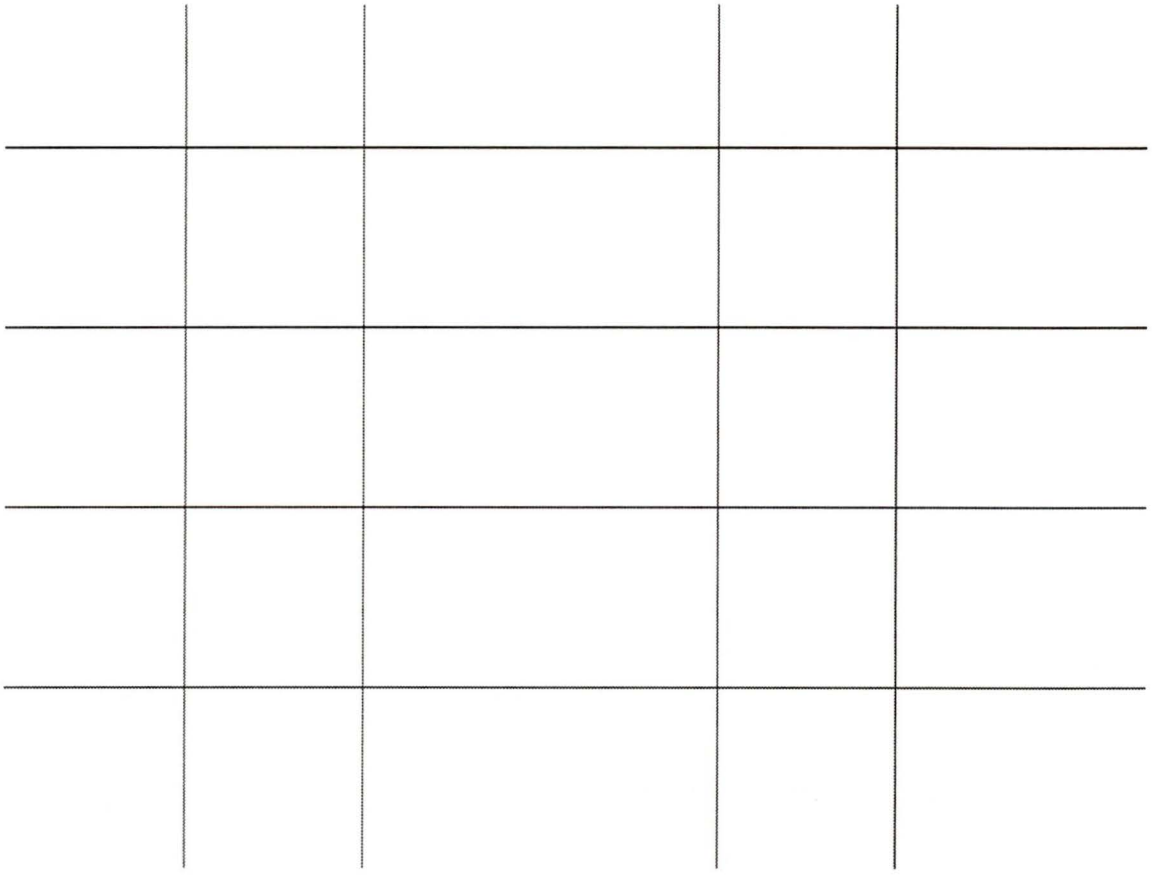

Figure 4.4: Synthetic scene.

close to the actual values used above. Correction of the lens distortion can be carried out using these parameters to warp the distorted image, resulting in the image shown in Figure 4.6, which closely approximates the original undistorted scene, as expected. The synthetic distortion has been almost completely removed.

The next chapter will discuss issues relating to the performance of this technique, including quantitative measures of of the deviation between the actual and computed warp parameters given above. Implementation issues inherent in the tools described above will also be discussed.

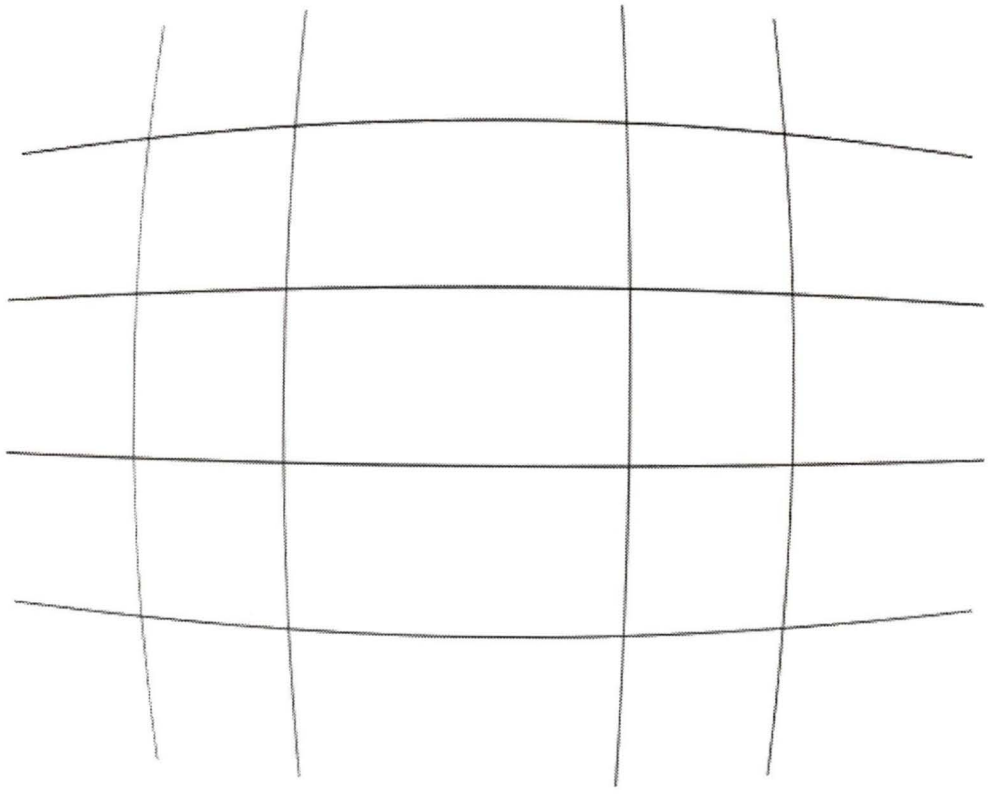

Figure 4.5: Synthetically distorted image.

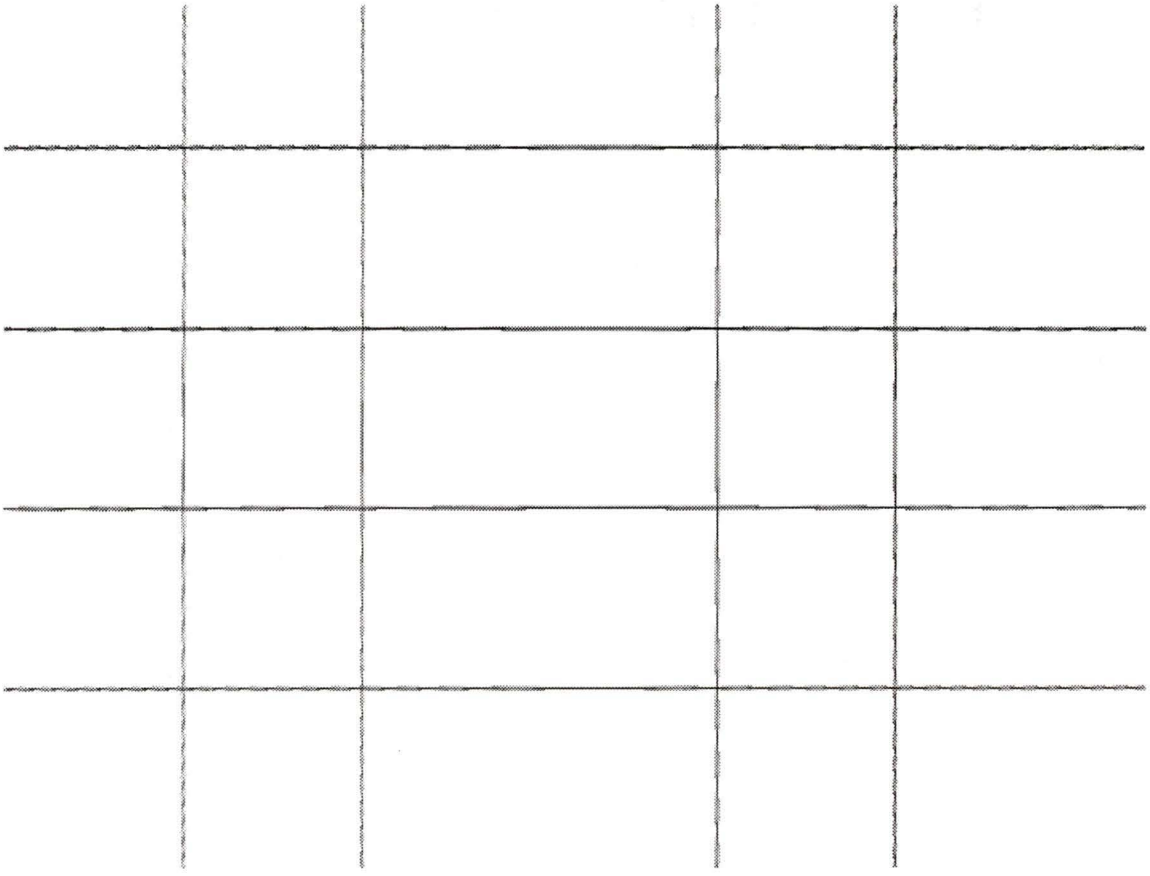

Figure 4.6: Corrected image.

Chapter 5

Experimental Results

This chapter describes the issues that were encountered while writing software to implement Automated Distortion Correction, and gives the results of testing that was carried out to validate the function of the software.

5.1 Implementation Issues

5.1.1 Programming Environment

MATLAB was chosen for development of the optimization algorithm. Version 4.1, running on a Sun workstation, has graphical output capabilities and a simple user interface which make it a very flexible environment for this application.

Subsidiary tools for image processing were written in the C language, using the Vision & Image Processing (VIP) package [10].

5.1.2 Point Selection

For best accuracy, many line support regions (LSRs), each containing many points, are used in the optimization of the distortion parameters. However, the distance between each such point and its associated line must be computed at each iteration of the optimization process. This means that more points result in longer computation time during the calibration. In order to shorten the computation time, a variety of sampling schemes have been used successfully to reduce the number of points used. Two were selected for quantitative analysis. Simplest of all is randomly sampling 10% of the points in the LSR, which works well. Even 10% of the full set of points gives good results. Another method which works well is to use a standard line detection algorithm, such as Canny's edge detector [8], as an additional test for each point. Points within each LSR which do not also exceed some Canny threshold are discarded. This leaves only single-pixel width lines with fewer points, which speeds up the optimization considerably, at a slight cost in the accuracy of the solution.

5.1.3 Scaling of Design Variables

MATLAB's general unconstrained minimization routine, `fminu`, uses the BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure, with derivatives calculated by finite difference approximation. The present version of `fminu` seems to have trouble with the wide discrepancy in the magnitudes of the design variables. They range in size from 10^{-16} for K_3 to 800 for ρ_m of some possible lines. The problem manifests itself when `fminu` jumps to a very distant point in the optimization space, where the cost function is huge, and then attempts to slowly work it's way back. To compensate for this effect, a set of fixed scale factors was used, which would change all the variables to be on the order of 1. These factors, plus scaling

of the cost function value to the same range, resulted in much better convergence on test problems.

5.1.4 Avoiding Local Minima

The next problem encountered had to do with MATLAB finding local minima of the objective function, instead of the desired global minimum. Usually, local minima were encountered when the true image centre, (c_x, c_y) migrated far from the centre of the image during the initial steps of the optimization, before the other warp parameters were even approximately known. Once this occurred, it was unlikely that the optimization would succeed without outside assistance.

The solution to this problem was to run the optimization in several steps. The first pass through the optimization process optimizes the variables $(\rho_m, \theta_m, K_1, c_x, c_y)$, ignoring the higher order K values, and using lax stopping conditions. This results in a quick first approximation for K_1, which accounts for the majority of the distortion in typical images. Subsequent optimizations add the K_2 and K_3 variables, if desired, the final pass using the desired stopping conditions to result in the needed precision of the result. The MATLAB function which implements this strategy is robust, and is able to locate the lines and warp parameters in synthetic test images to a high degree of accuracy, as demonstrated in the next section.

5.2 Performance Evaluation

Synthetic and real images have both been used to validate the performance of the algorithms. The advantage to synthetic images is that line positions, orientation, and distortion parameters are known exactly for the test images, so it is possible

to make precise quantitative evaluations of performance. However, dealing with the uncertainty inherent in real images is important as well, so some examples of real image correction are also shown in this section. In this case, several images taken with the same camera can be processed and the results compared, with the expectation that the same distortion parameters should be discovered.

5.2.1 Synthetic Image Tests

Two different warping functions were used on a synthetic grid image, simulating high and low distortion cameras, as shown in Figure 5.1. The distortion parameters used to create these images, and the results determined by the optimization process are shown in Table 5.1.

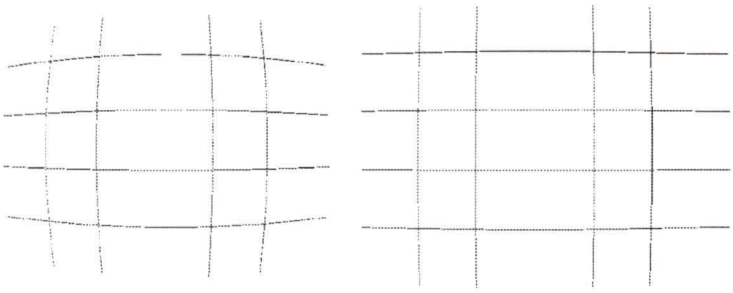

Figure 5.1: Synthetically warped images. Left: High distortion. Right: Low distortion.

The numbers in the table seem to be close together in most cases, but it is difficult to visualize the effects that these discrepancies would have on the corrected images. Table 5.2 and the following figures attempt to illustrate the effects of these discrepancies in a quantitative fashion. This is accomplished by computing the vectors corresponding to the distortion functions at each pixel in a 640 by 480 pixel

Table 5.1: Synthetic Image Distortion Experimental Results

	Image Centre		Distortion coefficients	
	c_x	c_y	$K_1(\times 10^{-7})$	$K_2(\times 10^{-12})$
High distortion (actual)	330.00	240.00	20.000	3.0000
High distortion (experimental)	331.17	239.44	20.081	2.8648
Low distortion (actual)	300.00	255.00	6.0000	-2.0000
Low distortion (experimental)	298.27	251.72	6.0343	-2.0118

image. Figures 5.2 and 5.3 show the magnitudes of these vectors for the high and low distortion images as heights of the surfaces. The surfaces are radially symmetric about the image centre, where the distortion is zero, and generally increase toward the edges of the image. A negative K_3 term can reverse this trend, however.

The difference between the actual and the experimentally determined distortion vectors at each pixel gives the residual distortion that remains after the correction function is applied to the distorted image. Effectively, these vectors measure the remaining distortion in the images after correction has been performed. The magnitudes of these residual distortion vectors are shown in Figures 5.4 and 5.5. Notice that the scales on the vertical axes of the graphs are much smaller than in the 'before' surfaces.

For perfect correction, all the residual vectors for each pixel would be zero, but the slight error accepted in the optimization process makes this impossible. Figures 5.6, 5.7, 5.8, and 5.9 show the distributions of the magnitudes of the distortion vectors, before and after correction for the high and low distortion images. Once again, notice that the distortion magnitudes on the horizontal axes of the 'after' histograms are much smaller.

Lastly, Table 5.2 shows the mean, median, and maximum lengths of the distortion vectors before and after correction, for each image. In both cases, the distortion has

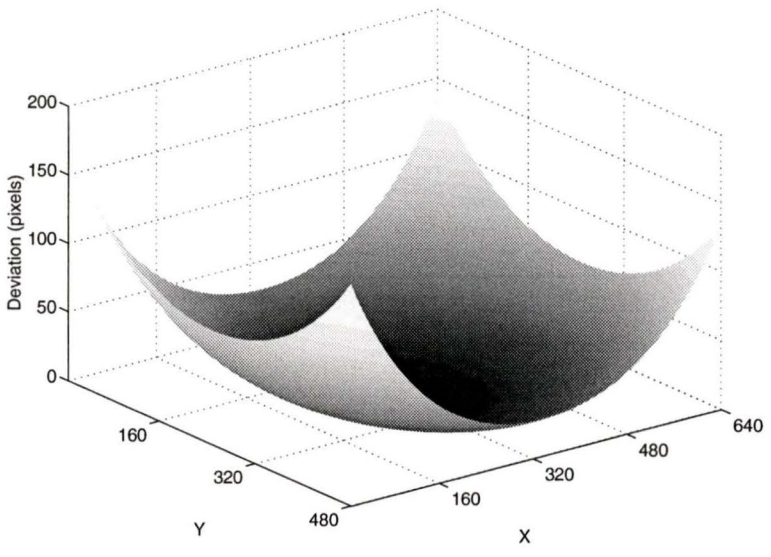

Figure 5.2: High distortion surface, before correction.

Table 5.2: Distortion Deviation Statistics, in Pixels

	Mean Error	Median Error	Maximum Error
High distortion (before)	32.4	23.2	169
High distortion (after)	0.367	0.282	2.34
Low distortion (before)	6.25	5.43	18.3
Low distortion (after)	0.149	0.147	0.360

been significantly reduced by the correction process, to the point where it would be undetectable in the output image for the low distortion case (the residual distortion is less than 0.5 pixels over the entire image). Even for the high distortion case, the maximum distortion has been reduced by about 98.6%.

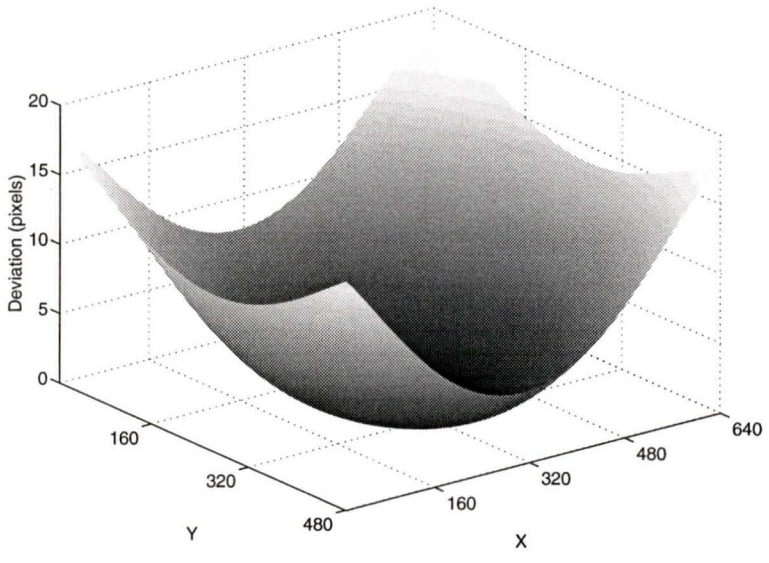

Figure 5.3: Low distortion surface, before correction.

5.2.2 Real Image Tests

Blind Images

A series of four images of a set of horizontal window blinds were taken using an inexpensive 640 by 480 pixel CCD camera. Two images were taken with the camera in a horizontal orientation, and two with the camera in a vertical orientation. Care was taken to arrange the camera and lighting of the scene to minimize the occurrence of spurious lines due to other objects and shadows, leaving only the strong lines due to the slats of the blinds. Figures 5.10 and 5.11 show one each of the horizontal and vertical blind images.

After detecting linear features in the blind images using the method described in section 4.1, points from the two horizontal and two vertical line images were combined into two independent data sets. Each of these sets was then sampled in two ways:

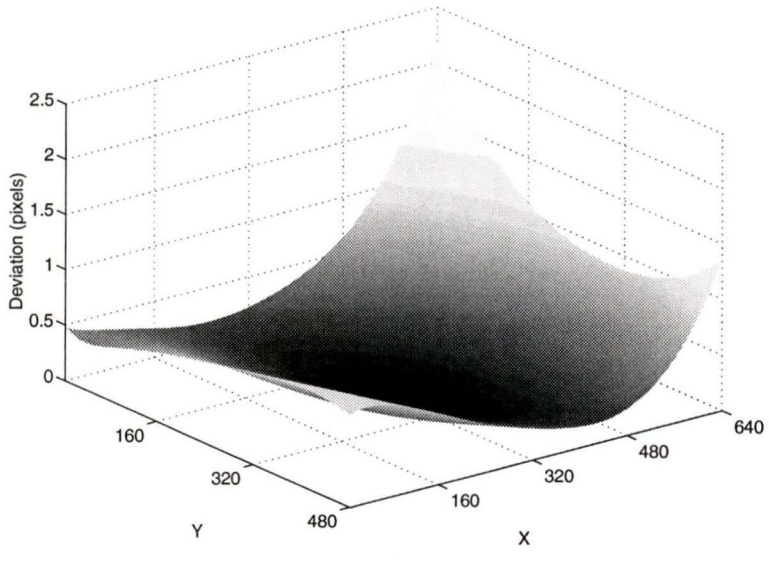

Figure 5.4: High distortion surface, after correction.

- 10% random sampling
- Canny sampling followed by 10% random sampling

Figure 5.12 shows an example of one of the four resulting sets of points. Each of these sets was then used as data for the optimization process, resulting in the distortion parameters shown in Table 5.3.

Table 5.3: Blind Image Distortion Experimental Results

	Image Centre		Distortion coefficients	
	c_x	c_y	$K_1(\times 10^{-7})$	$K_2(\times 10^{-12})$
Set 1	334.9	264.3	5.90	-1.25
Set 2	332.2	265.5	5.87	-1.16
Set 1 (Canny)	335.0	267.7	6.43	-1.49
Set 2 (Canny)	336.1	266.3	6.69	-1.60

Since both data sets came from the same camera, at identical camera settings,

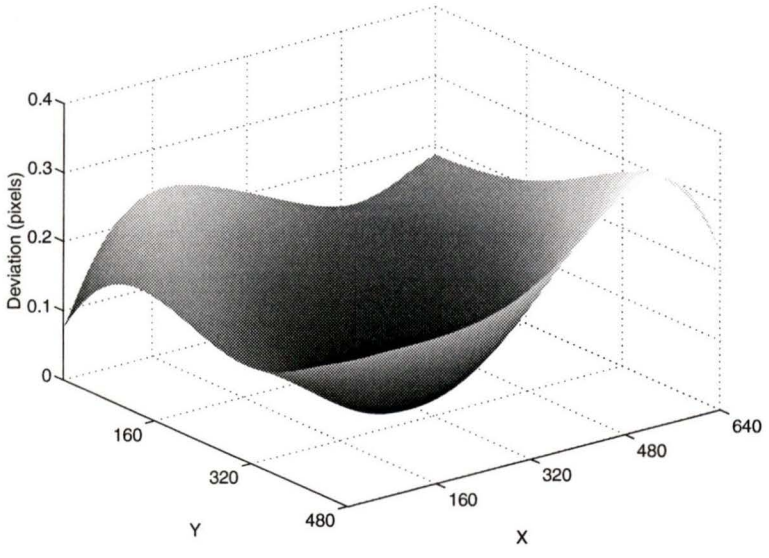

Figure 5.5: Low distortion surface, after correction.

the distortion parameters should be similar. However, since the actual values are not known, as they were for the synthetic images, no absolute error analysis is possible. Instead, the results from the data sets can only be compared to each other, in order to give an indication of the consistency of the values.

Table 5.4 gives the mean, median, and maximum distances, in pixels, between the new locations of image pixels corrected using combinations of the distortion parameters shown in Table 5.3.

Using the larger 10% sampled data set results in the minimum mean uncertainty of 0.17 pixels, but all the mean errors are in the neighbourhood of 0.5 pixels, with the maximum difference around 2 pixels. For comparison, the last line of the table gives the mean, median, and maximum distortion corrected by the warping functions derived from Set 1, which are estimates of the initial distortion in the image.

The final result of this process is to use one of the derived warp functions to

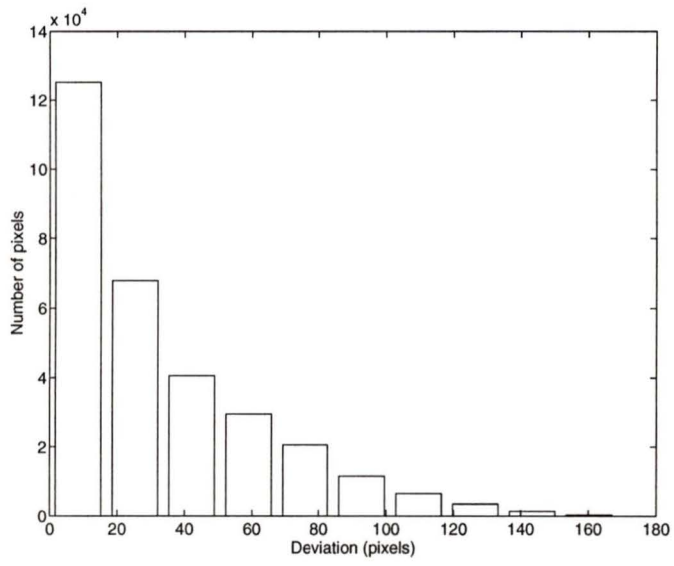

Figure 5.6: High distortion histogram, before correction.

correct one of the original input images. Figure 5.13 shows this result. By applying the warp parameters determined by Set 1 in Table 5.3 to the image in Figure 5.10, the resulting image exhibits much less distortion, which can be seen in the straightness of the lines, especially near the edges of the image.

Low Distortion Camera Images

A second series of images was also taken for analysis, using a more sophisticated camera, on loan from Eos Systems, of Vancouver. This camera had a higher resolution, 1524 by 1012 pixels, and low distortion lenses. Two sets of images were taken, one of a set of window blinds, and one of a garage door with a linear texture. Each set consisted of two images, one taken with the camera oriented vertically and one with the the camera oriented horizontally. Figures 5.14 and 5.15 are examples.

These images were processed using Canny sampling, as described in the previous

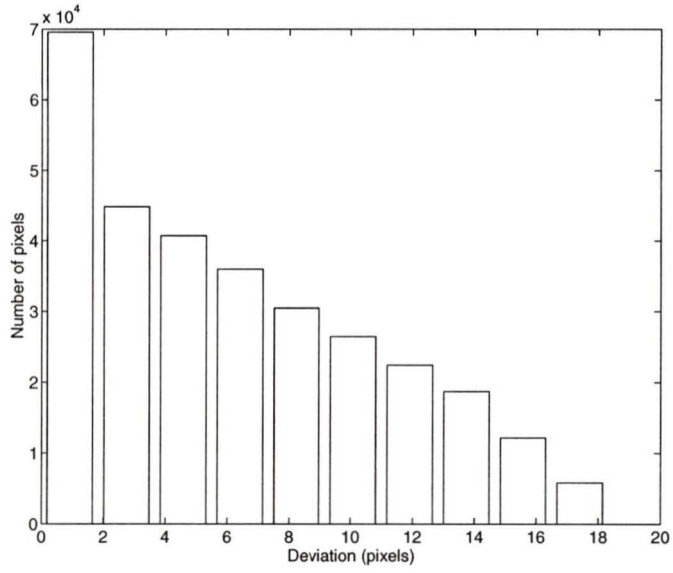

Figure 5.7: Low distortion histogram, before correction.

section. Table 5.5 gives the results of the optimization process. Notice that the distortion coefficients are smaller than in Table 5.3. The K_2 values in particular are very small, and are not consistent between trials for that reason.

Table 5.6 reflects the close agreement between the two sets of warp parameters. The bottom line again shows the magnitude of the distortion, for comparison. Finally, Figure 5.16 is the image that results from applying the warp function shown in the first line of Table 5.5 to the image shown in Figure 5.14. The improvement in the distortion is too small to be detected visually; however, for quantitative analyses, a two-pixel correction may be very important.

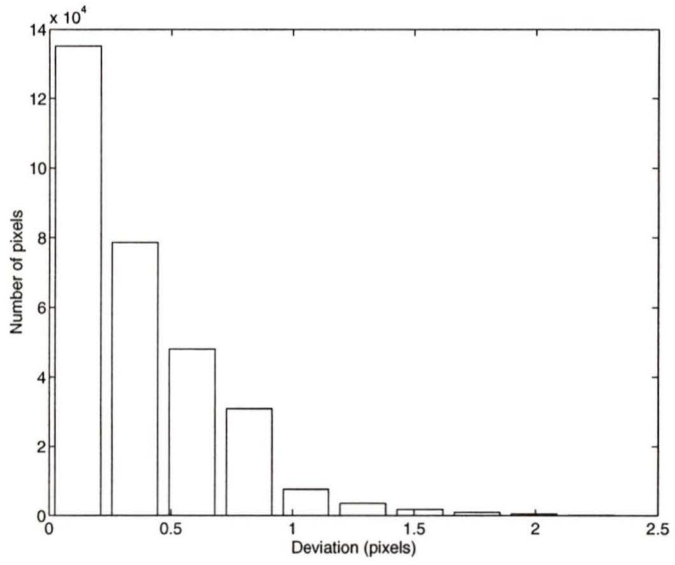

Figure 5.8: High distortion histogram, after correction.

5.3 Performance Conclusions

This chapter has presented a number of examples of the use of the Automated Distortion Correction algorithm on both real and synthetic image data. In all cases, the processing of the images ended successfully in the determination of a set of distortion parameters which could be used to correct the input images to remove the radial distortion inherent in them.

Using synthetic images, it has been shown that the distortion parameters recovered by this algorithm can correct images to within 0.4 pixels average deviation of the actual distortion in the images, with the maximum distortion being reduced by up to 98.6%.

Real images were also used to demonstrate that for high and low distortion lenses, a significant decrease in radial distortion can be achieved by use of this algorithm.

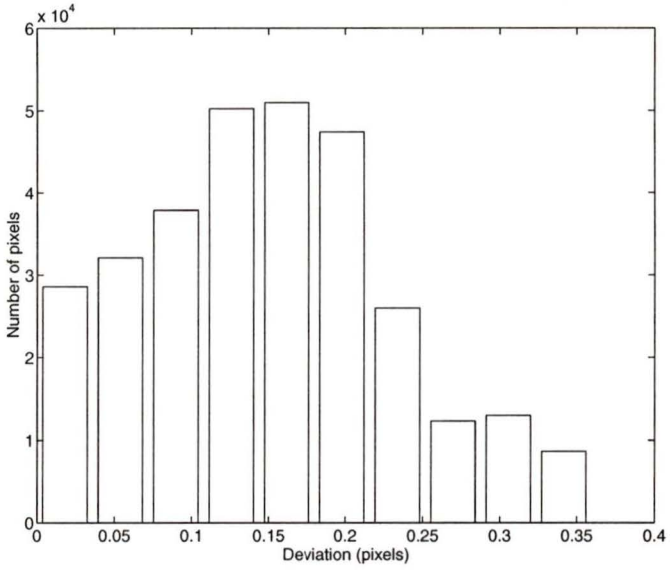

Figure 5.9: Low distortion histogram, after correction.

Table 5.4: Blind Distortion Deviation Statistics, in Pixels

Comparison	Mean	Median	Maximum
Set 1, Set 2	0.17	0.13	1.15
Set 1 (Canny), Set 2 (Canny)	0.26	0.21	0.88
Set 1, Set 1 (Canny)	0.51	0.47	1.34
Set 2, Set 2 (Canny)	0.69	0.62	1.78
Set 1, Set 2 (Canny)	0.76	0.69	1.94
Set 2, Set 1 (Canny)	0.44	0.36	1.15
Set 1, -	7.1	5.7	28.0

Table 5.5: Low Distortion Image Experimental Results

	Image Centre		Distortion coefficients	
	c_x	c_y	$K_1(\times 10^{-7})$	$K_2(\times 10^{-12})$
Set 1 (Blinds)	761.6	513.0	0.140	-0.00103
Set 2 (Door)	758.3	503.8	0.129	0.00036

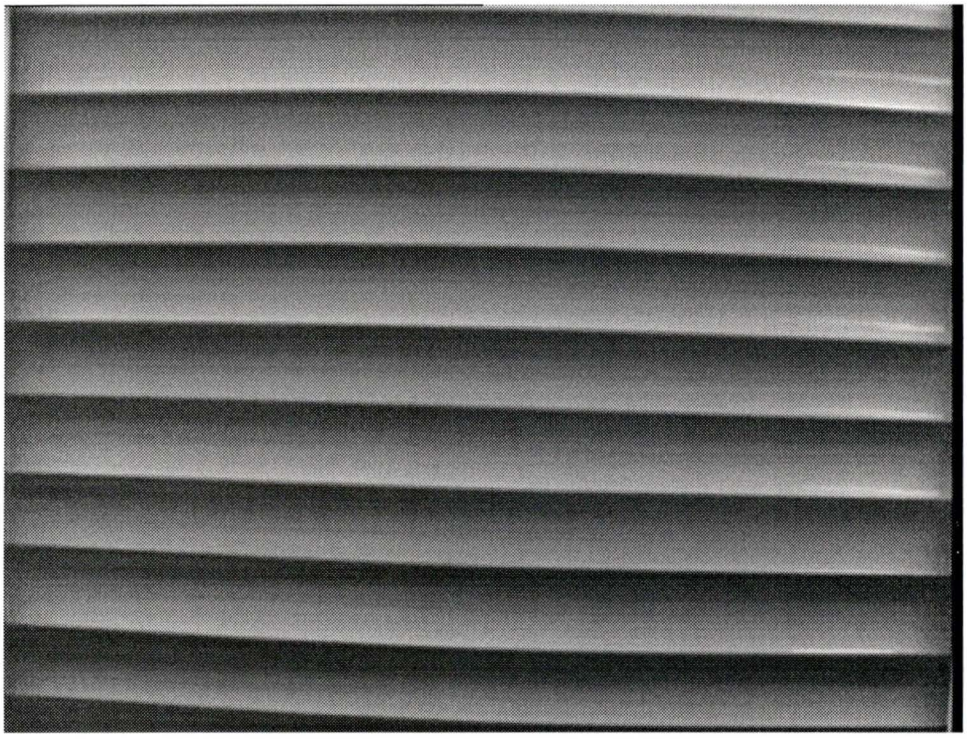

Figure 5.10: Horizontal line image of blinds.

Table 5.6: Low Distortion Deviation Statistics, in Pixels

Comparison	Mean	Median	Maximum
Set 1, Set 2	0.10	0.08	0.35
Set 1	2.34	1.67	10.19

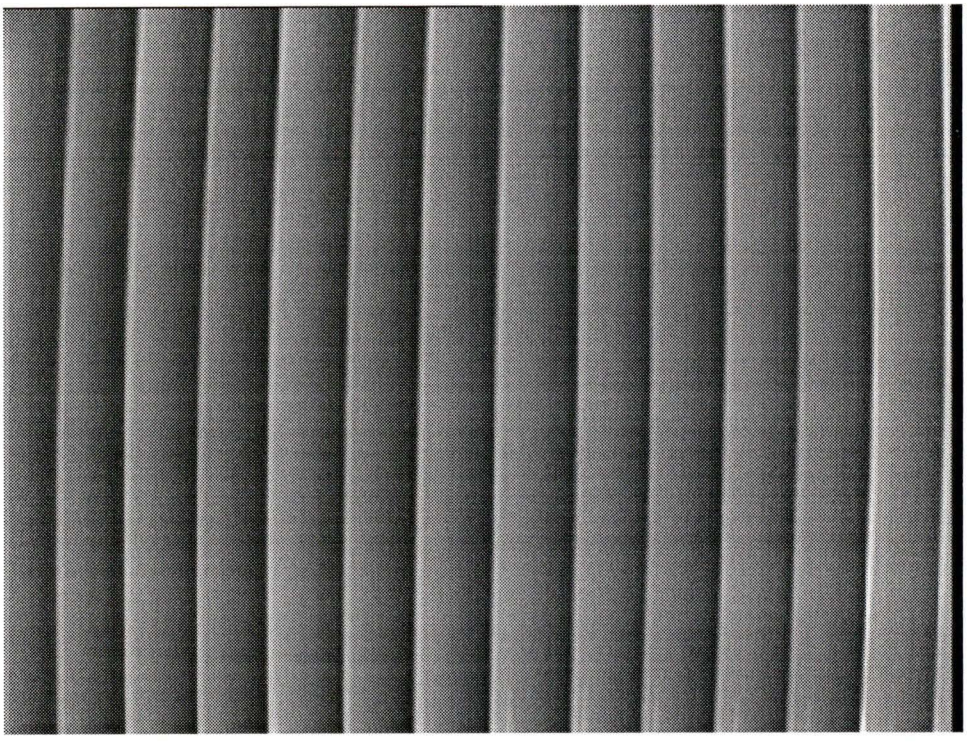

Figure 5.11: Vertical line image of blinds.

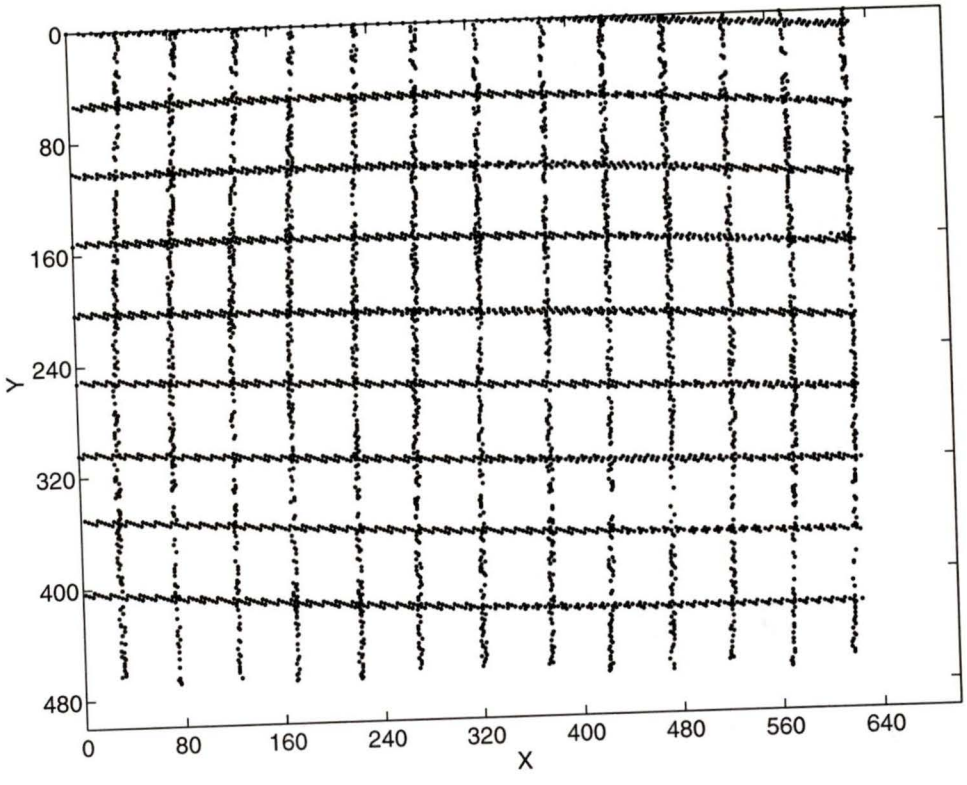

Figure 5.12: Blind images combined to form grid.

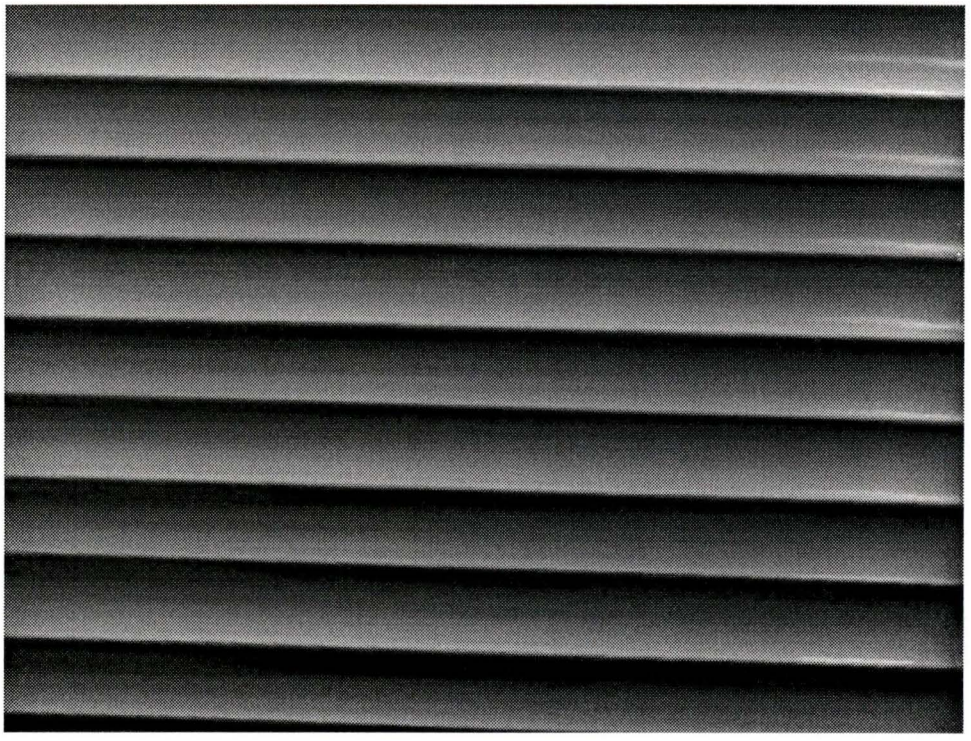

Figure 5.13: Corrected image of blinds.

Figure 5.14: Low distortion image of blinds.

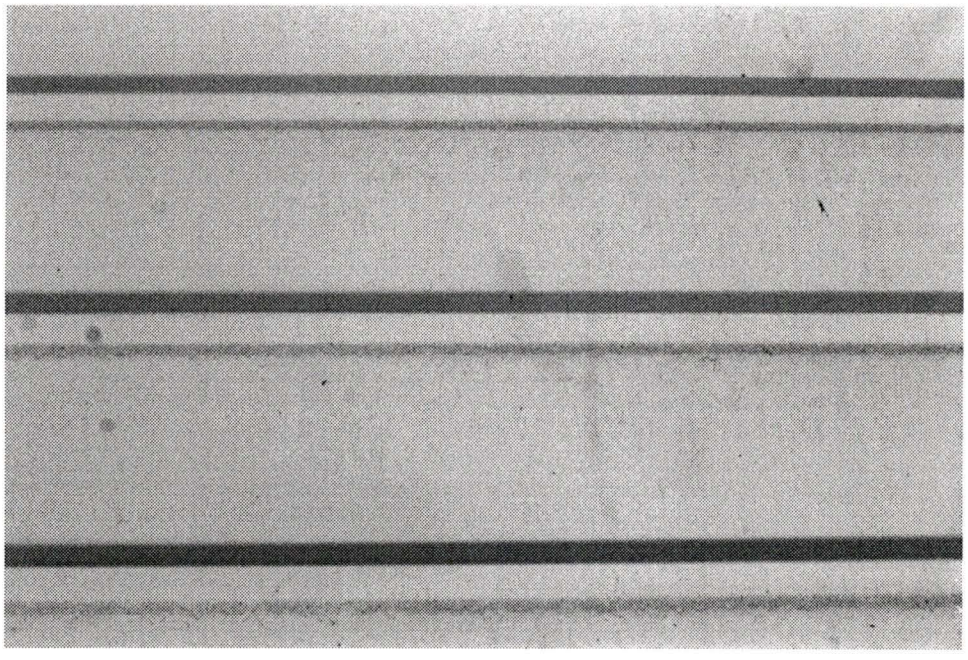

Figure 5.15: Low distortion image of garage door.

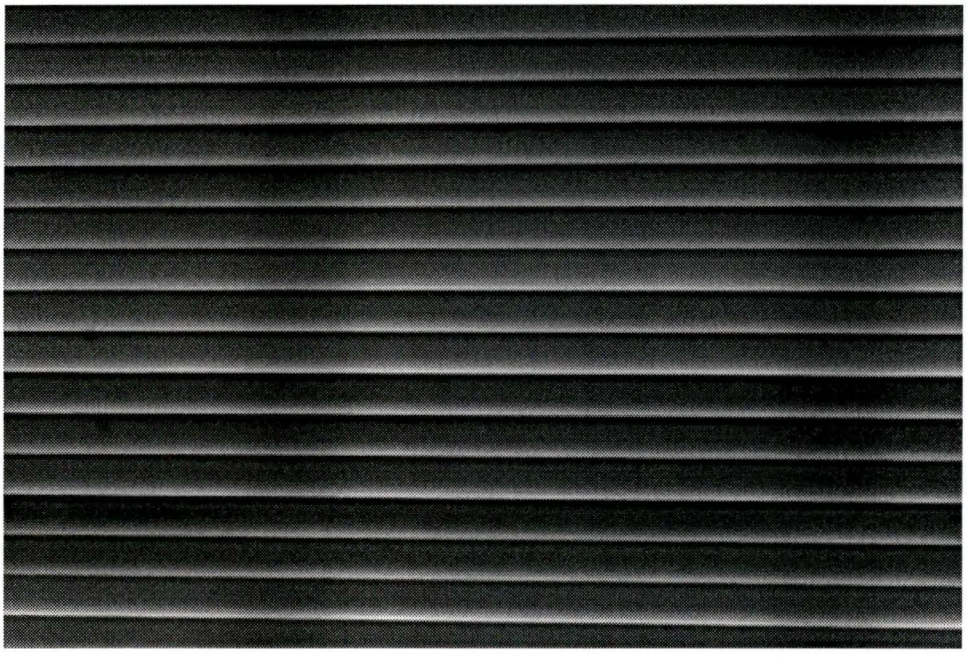

Figure 5.16: Corrected image of blinds.

Chapter 6

Conclusions and Future Work

The algorithm described in this thesis is successful at determining the optimal warping function for the correction of radially distorted images. Calibration images can be taken of simple, readily available scenes, with no measurement of calibration objects required. These images are then analysed, and through optimization, an appropriate warping function can be determined to compensate for any radial distortion. This warping function allows either entire images or individual feature points to be efficiently corrected, eliminating the radial distortion.

6.1 Shortcomings

The principal shortcoming of the described procedure is that the calibration phase requires large amounts of computing time and power. Time constraints make it less attractive to re-calibrate a camera each time the focal length or aperture are changed. Ideally, calibration should require only a few minutes, instead of the current 10 to 30 minutes. This shortcoming will decrease as the available computing power increases.

Also, it would be an improvement if the calibration procedure included a better model for distortion, taking into account decentring distortion. This would result in a slight increase in accuracy, but may not justify the additional computing power.

6.2 Future Work

In the future, the described algorithm for the correction of radial distortion will be used as a starting point for more advanced image processing applications. First and foremost will be a complete camera calibration algorithm, which will be based upon images of parallel lines and vanishing points. As well, work on correction of art images for cataloguing and restoration is planned. An outside company has also expressed an interest in making use of the algorithm as a way to improve the accuracy of a robotic visual tracking system.

Bibliography

[1] D.H. Ballard and C.H. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, N.J., 1982.

[2] Thaddeus Beier. Feature-based image metamorphosis. *Computer Graphics*, 26(2):35–42, July 1992.

[3] Eric A. Bier and Jr. Kenneth R. Sloan. Two-part texture mappings. *IEEE Computer Graphics and Applications*, pages 40–53, September 1986.

[4] S. Boukharouba, J.M. Rebordao, and P.L. Wendel. An amplitude segmentation method based on the distribution function of an image. *Comput. Vision Graphics and Image Proc.*, 29:47–59, 1985.

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

[6] Duane C. Brown. Calibration of close range cameras. In *Proceedings of the XII Congress of the International Society of Photogrammetry*, pages 1–26, Ottawa, Canada, July 23 through August 5 1972.

[7] J.B. Burns, A.R. Hansen, and E.M. Riseman. Extracting straight lines. *IEEE Trans. Pattern Analysis Machine Intelligence*, PAMI-8:425–455, july 1986.

[8] J. F. Canny. A computational approach to edge detection. *IEEE Pattern Analysis and Machine Intelligence*, (PAMI-8):679–698, 1986.

[9] John Cardillo and Maher A. Sid-Ahmed. 3-D position sensing using a passive monocular vision system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):809–812, August 1991.

[10] R. D. Dony, G. F. McLean, and M. E. Jernigan. VIP: The vision and image processing program. *IEEE Conference on Systems, Man and Cybernetics*, 1987.

[11] Morris R. Driels and Uday S. Pathre. Visions-based automatic theodolite for robot calibration. *IEEE Transactions on Robotics and Automation*, 7(3):351–360, June 1991.

[12] Tomio Echigo. A camera calibration technique using three sets of parallel lines. *Machine Vision and Applications*, 1(3):159–167, 1990.

[13] S. F. El-Hakim. Real-time image metrology with CCD cameras. *Photogrammetric Engineering and Remote Sensing*, 52(11):1757–1766, November 1986.

[14] Jan Flusser. An adaptive method for image registration. *Pattern Recognition*, 25(1):45–54, 1992.

[15] Ardeshir Goshtasby. Piecewise cubic mapping functions for image registration. *Pattern Recognition*, 20(5):525–533, 1987.

[16] Ardeshir Goshtasby. Image registration by local approximation methods. *Image and Vision Computing*, 6(4):255–261, November 1988.

[17] Ardeshir Goshtasby. Registration of images with geometric distortions. *IEEE Transactions on Geoscience and Remote Sensing*, 26(1):60–64, January 1988.

[18] Ardeshir Goshtasby, George C. Stockman, and Carl V. Page. A region-based approach to digital image registration with subpixel accuracy. *IEEE Transactions on Geoscience and Remote Sensing*, GE-24(3):390–399, May 1986.

[19] William I. Grosky and Louis A. Tamburino. A unified approach to the linear camera calibration problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):663–671, July 1990.

[20] Valerie Hall. Introduction to morphing. Report for computing topics 591, Curtin University of Technology, Perth, Western Australia, July 1992.

[21] R.M Haralick and L.G. Shapiro. *Computer and Robot Vision.* Addison Wesley, Reading, Mass., 1992.

[22] Paul S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, November 1986.

[23] Gerald Q. Maguire Jr., Marilyn E. Noz, Henry Rusinek, Judith Jaeger, Elissa L. Kramer, Joseph J. Sanger, and Gwenn Smith. Graphics applied to medical image restoration. *IEEE Computer Graphics and Applications*, pages 20–28, March 1991.

[24] Reimar Lenz and Dieter Fritsch. Accuracy of videometry with CCD sensors. *ISPRS Journal of Photogrammetry and Remote Sensing*, (45):90–110, 1990.

[25] Reimar K. Lenz and Roger Y. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):713–720, September 1988.

[26] Z. C. Li, C. Y. Suen, T. D. Bui, Y. Y. Tang, and Q. L. Gu. Splitting-integrating method for normalizing images by inverse transformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6):678–686, June 1992.

[27] Y. Liu and T.S. Huang. Determining straight line correspondences from intensity images. *Pattern Recognition*, 24(6):489–504, 1991.

[28] G.F. McLean and D. Kotturi. Detection of vanishing points by line clustering. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 1993.

[29] G.F. McLean, B. Prescott, and D. Kotturi. Hierarchical clustering for automated line detection. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 244–247, 1993.

[30] Mehran Moshfeghi. Elastic matching of multimodality medical images. *Computer Vision, Graphics, and Image Processing*, 53(3):271–282, May 1991.

[31] Dana Norvila and Jamie Dixon. Computer-generated metamorphoses evolve. *IEEE Spectrum*, page 11, April 1992.

[32] P.K. Sahoo, S. Soltani, A.K. Wong, and W.C. Chen. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233–260, 1988.

[33] M.I. Sezan. A peak detection algorithm and its application to histogram-based image data reduction. *Comput. Vision Graphics and Image Proc.*, 49:36–51, 1990.

[34] Mark R. Shortis. Precision evaluations of digital imagery for close-range photogrammetric applications. *Photogrammetric Engineering and Remote Sensing*, 54(10):1395–1401, October 1988.

[35] M.A. Sid-Ahmed and Mohamed T. Boraie. Dual camera calibration for 3-D machine vision metrology. *IEEE Transactions on Instrumentation and Measurement*, 39(3):512–516, June 1990.

[36] Roger Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. *CVGIP*, pages 364–374, 1986.

[37] Roger 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.

[38] Ling-Ling Wang and Wen-Hsiang Tsai. Camera calibration by vanishing lines for 3-D computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):370–376, April 1991.

[39] Guo-Qing Wei and Song De Ma. A complete two-plane calibration method and experimental comparisons. *1993 Proceeding of the International Conference on Computer Vision*, pages 439–446, 1993.

[40] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, October 1992.

[41] George Wolberg. *Digital Image Warping.* IEEE Computer Society Press, Washington, 1990.

VITA

Surname: Prescott Given Names: Blair

Place of Birth: Surrey, British Columbia, Canada Date of Birth: November 1, 1969.

Educational Institutions Attended:

University of Victoria	1992 to 1994
University of British Columbia	1987 to 1992

Degrees Awarded:

B.A.Sc., University of British Columbia	1992

Honours and Awards:

Charles and Jane Banks Foundation Scholarship	1989
University of British Columbia Fellowship	1988
Canadian Army Remembrance Scholarship	1987
Borden Chemical Canada Scholarship	1987
Yates Memorial Scholarship	1987
Constable Tom Agar Memorial Scholarship	1987
Grade 12 Provincial Scholarship	1987

Publications:

G. F. McLean and B. Prescott, 1991. *Teleoperator Task Performance Evaluation Using 3-D Video.* IEEE Pacific Rim Conference on Comunications, Computers and Signal Processing, Victoria, May 1991, pp. 615–619.

Blair Prescott and G. F. McLean, 1994. *Performance Evaluation Measures for Operator-Telerobot Systems.* IEEE Transactions on Systems, Man and Cybernetics, May/June 1994.

G. F. McLean, B. Prescott and D. Kotturi, 1993. *Hierarchical Clustering for Automated Line Detection.* IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing, pp. 244–247, 1993.

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis:

Automated Calibration and Correction of Radial Lens Distortion

Author: [redacted]
Blair Prescott

Sept 28, 1994.
Date