

**AUTOMATIC RADIAL DISTORTION ESTIMATION FROM A
SINGLE IMAGE**

by

Faisal Bukhari

A dissertation submitted in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in
Computer Science

Examination Committee: Dr. Matthew N. Dailey (Chairperson)
Dr. Nitin V. Afzulpurkar (Member)
Dr. Raphaël Duboz (Member)

External Examiner: Dr. Luis Gómez Déniz
Centro de Tecnologías de la Imagen (CTIM),
Universidad de Las Palmas de Gran Canaria (ULPGC),
Spain

Nationality: Pakistani
Previous Degree: Masters of Science in Computer Science
Asian Institute of Technology, Thailand

Scholarship Donor: Higher Education Commission (HEC), Pakistan - AIT
Fellowship

Asian Institute of Technology
School of Engineering and Technology
Thailand
November 1, 2012

Dedication

This thesis is dedicated to my mother, Naheed Akhtar and my wife, Syeda Fozia Bukhari for their love, endless support, understanding and encouragement.

Acknowledgments

I would like to thank ALLAH, THE ALMIGHTY for always being with me in every aspect of my life.

I would also like to acknowledge the contributions and encouragements of my advisor, committee members, colleagues, friends, and family members. This encouragement has ultimately resulted in the completion of this dissertation.

First, I would like to thank my mentor, Matthew N. Dailey, for his rigorous feedback, continued support, and keen interest in my research. He always motivated and guided me not only in my research but also in other aspects of life. His concrete feedback and rational thinking always helped me to stay focused in my research. The lab scrums provided me an opportunity to get his feedback thrice a week. I am also thankful to Nitin V. Afzulpurkar and Raphaël Duboz for their useful comments and feedback during my presentations.

Second, I am very grateful to my external examiner Dr. Luis Gómez Déniz, for his insightful comments and valuable feedback.

Third, I would like to thank the eminent Pakistani scientist, Prof. Dr. Atta ur Rehman, who is the founder of Higher Education Commission (HEC) of Pakistan. It is due to his vision that thousands of Pakistani students have been given scholarships to study overseas. This critical step is helping Pakistan to improve its research productivity and increase quality standards for university faculty.

Fourth, I would like to thank my donor agency, the Higher Education Commission (HEC) of Pakistan for providing me the scholarship for the M.S. leading to Ph.D. program at AIT and for funding me to present my work at an international conference.

Fifth, it has been an honor for me to work in the Vision and Graphics Lab (VGL) headed by Matthew. The environment in the lab is excellent and conducive to research. The mutual cooperation among the members is exemplary. I would like to thank my colleagues in the VGL, especially Irshad Ali, Waheed Iqbal, Waheed Noor, Kifayat Ullah, Abdul Basit, Waqar Shahid, Jednipat Moonrinta, Zia Uddin, Sher Doudpota, Sergio Goncalves, Supawadee Chaivivatrakul, and Kan Ouivirach, for their valuable feedback and support throughout my research at AIT.

Sixth, I would like to thank Mr. Olivier Christian Nicole, laboratory and research manager at Computer Science and Information Management (CSIM) for providing excellent services to all students.

Seventh, I express my sincere gratitude to my late father, Syed Sajid ur Rehman Bukhari and my late uncle, Muhammad Yameen Farooqi for their endless love, support, motivation, and guidance. May Allah grant their souls eternal peace. AMEEN.

Eight, I express my love and gratitude to my beloved mother, Nahid Akhtar, whose continued encouragement and infinite support has guided me a great deal in setting my goals. He has been a great source of motivation during difficult times in my life. Also I would like to praise my brothers Aamir, Khurram, Saud, Yasir, and Nasir, and my sister Tehmina for their support, encouragement, love, and prayers.

Last but not least, my wife, Fozia, has always supported, motivated, and took keen interest in my research over the years. She is my most trusted friend and a personal counselor. She has taken great care of our home, whenever I was busy in my research and spent most of her time with our kids. I must admit that without her help I could not have finished my dissertation in time. I would also like to praise our children Abdullah, Rabbiya, and Zarar for their love, prayers, and wonderful activities that have always helped me relax and freshen my mind.

Abstract

Many computer vision algorithms rely on the assumptions of the pinhole camera model, but lens distortion with off-the-shelf cameras is usually significant enough to violate this assumption. Many methods for radial distortion estimation have been proposed, but they all have limitations. Robust automatic radial distortion estimation from a single natural image would be extremely useful for many applications, particularly those in human-made environments containing abundant lines. For example, it could be used in place of an extensive calibration procedure to get a mobile robot or quadrotor experiment up and running quickly in an indoor environment.

In this dissertation we propose a new and fully automatic method for radial distortion estimation based on the plumb-line approach.

First, the method works from a single image and does not require a special calibration pattern. It is based on Fitzgibbon's division model.

Second, we devise a new algorithm for robust estimation of circular arcs.

Third, we design and implement a new algorithm for robust estimation of lens distortion parameters based on the estimated circular arcs.

Fourth, we perform an extensive empirical study of the method on synthetic images. We develop our own data set for synthetic images under different levels of lambda and distortion center.

Fifth, we perform a comparative statistical analysis of how different circle fitting methods contribute to accurate distortion parameter estimation.

Sixth, we provide qualitative results on a wide variety of challenging real images. The experiments demonstrate the method's ability to accurately identify distortion parameters and remove distortion from images. Seventh, we perform a direct comparison of our method with that of Alvarez et al. (Alvarez, Gomez, & Sendra, 2009), the only researchers who have provided a publicly accessible implementation of their method, on synthetic images.

Finally, we provide the source code based on OpenCV (Bradski, 2000) online¹ for researchers interested in evaluating or extending our procedure.

¹See <http://www.cs.ait.ac.th/vgl/faisal/downloads.html>.

Table of Contents

Chapter	Title	Page
	Title Page	i
	Acknowledgments	iii
	Abstract	v
	Table of Contents	vi
	List of Figures	vii
	List of Tables	xi
1	Introduction	1
	1.1 Motivation	1
	1.2 Contributions	2
	1.3 Dissertation Outline	3
2	Literature Review	4
	2.1 Point correspondence based methods	4
	2.2 Multiple view auto-calibration based methods	8
	2.3 Plumb-line methods	16
	2.4 Analysis of state of the art of plumb-line based methods	27
	2.5 Conclusion	27
	2.6 The Canny edge detector	28
	2.7 Random Sample and Consensus (RANSAC)	28
	2.8 Circle fitting methods	29
	2.9 Algebraic circle fitting methods	29
	2.10 Geometric circle fitting methods	31
3	Methodology	35
	3.1 Mathematical Model	35
	3.2 Robust Radial Distortion Estimation	40
4	Experiments and Results	45
	4.1 Quantitative Results on Synthetic Images	45
	4.2 Qualitative Results on Real Images	48
	4.3 Failure case analysis	65
	4.4 Comparison with Alvarez et al. (Alvarez et al., 2009) method	65
5	Conclusion	68
	References	69

List of Figures

Figure	Title	Page
2.1	Two calibration patterns. (a) Calibration pattern in 2D. (b) Calibration pattern in 3D. Reprinted from Moons, Gool, and Vergauwen (2009).	5
2.2	Schematic diagram for camera calibration. Reprinted from Tsai (1987).	6
2.3	Two sets of images of a calibration pattern. Images (a), (c), and (e) are taken from closer distances. Images (b), (d), and (f) are taken from further distances. Reprinted from Zhang (2000).	7
2.4	Two calibration patterns. (a) Grid pattern. (b) Point pattern. Reprinted from Braüer-Burchardt (2004).	8
2.5	Development of dual curve. Suppose p_1 and p'_1 denote the two epipoles for the first motion of the camera. The epipolar transformation is based on the Steiner conic s_1 passing through p_1 and p'_1 . If y is a point of s_1 , then two epipolar lines $\langle p_1, y \rangle$ and $\langle p'_1, y \rangle$ correspond to each other. The two tangents from p_1 to w intersect s_1 at points x_1, x_2 . The line segment $\langle x_1, x_2 \rangle$ of s_1 corresponds to the point $x_1 \times x_2$ in the dual of the image plane. The point $x_1 \times x_2$ lies on a curve g that is an algebraic transformation of c . To find the camera calibration uniquely, three camera displacements which give six conditions are enough. Note that Steiner theorem states that there is always a distinct conic s_1 passing through p_1 and p'_1 . y is a point of s_1 , if and only if, $\langle p_1, y \rangle \perp \langle p'_1, y \rangle$. Reprinted from Faugeras, Luong, and Maybank (1992).	9
2.6	Calibration images. Images (a), (b), and (c) are used for calibration. Image (d) is taken with a longer focal length, and lines are curved. Reprinted from Stein (1996).	10
2.7	Stereo images. Images (a) and (b) represent a stereo pair. (c) Circles represent points in one image, and lines represent their tentative corresponding points in the other image. RANSAC is used for input matches. The different color channels are superimposed in the images. Reprinted from Fitzgibbon (2001).	12
2.8	Hartley and Kang (2007)'s method sample results. (a) Checkerboard. (b) Detected grid. (c, d) Distorted images. (e, f) Corresponding undistorted images. Reprinted from Hartley and Kang (2007).	14
2.9	Non-planar calibration grid. Reprinted from Hartley and Kang (2005).	15
2.10	Flow curves. (a) Pure rotations (b) Pure translation. (c) Intersection of rotation and translation flow curves. Reprinted from Ramalingam, Sturm, and Lodha (2010).	15

2.11	Example of image sequence. (a-d) Distorted images . (e-h) Undistorted images. Reprinted from Kukelova and Pajdla (2011).	16
2.12	A close-range camera. The camera accepts 190×215 mm by $\frac{1}{4}$ inch ballistic plates. Also it exposes a 6-inch diameter format. Focusing can be done from 2 feet to infinity using interchangeable cones. Reprinted from Brown (1971).	17
2.13	Synthetic images with no noise. (a) Original image. (b) Manually selected points. (c) Undistorted image. (d) Manually drawn approximate lines. (e) Results from conventional snake algorithm. (f) Results from radial distortion snake algorithm. (g) Undistorted images (e) and (f). Reprinted from Kang (1997).	18
2.14	Synthetic images with Gaussian noise with standard deviation of 100. (a) Original image. (b) Manually drawn approximate lines. (c) Results of conventional snake algorithm. (d) Results of radial distortion snake algorithm. Reprinted from Kang (1997).	19
2.15	Manual method is based on manual selection of points that are used to compute the radial distortion parameters. (a) Original image. (b) Manually drawn lines. (c) Undistorted image. Reprinted from Kang (1997).	20
2.16	Conventional snake method allows user to draw snake contour on the image that is equivalent to a line in the world. (a) Chosen snakes using conventional snake method. (b) Optimized snake contour. (c) Undistorted image. Reprinted from Kang (1997).	20
2.17	Radial distortion snake method is based on independent and conventional snake contour. (a) Chosen snakes using radial distortion snake method. (b) Optimized snake contour. (c) Undistorted image. Reprinted from Kang (1997).	20
2.18	Comparing the two snake implementation. (a) Initial and optimized snake contour for comparing the two snake algorithms. (b) Conventional snake. (c) Radial distortion snake. Reprinted from Kang (1997).	21
2.19	Radial distortion removal example. (a) Initial hand-drawn snake on black line super imposed on column corner at left side of image. (b) Optimized snake contour. (c) Undistorted image. Reprinted from Kang (2000).	21
2.20	Radial distortion removal example based on Tsai (1987)'s image. (a) Distorted image. (b) Undistorted image. Reprinted from Devernay and Faugeras (2001).	22
2.21	Thormählen, Broszio, and Wassermann (2003)'s procedure for the undistorted lines. (a) Detection of points p_d on curved lines. (b) From segments of curved lines, linkage of straight lines. (c) Points p_d on the curved line are mapped to points p_u on the associated straight line Λ_m . It is a function of k_3 , k_5 and the points p_d . Reprinted from Thormählen et al. (2003).	22
2.22	Undistorted image using El-Melegy and Farag (2003)'s method. (a) Distorted image. (b) Undistorted image. Reprinted from El-Melegy and Farag (2003).	23

2.23	Sample results of Strand and Hayman (2005)'s method. (a) Original image. (b) Undistorted image. (c) Figures of line before and after undistortion. Reprinted from Strand and Hayman (2005).	23
2.24	Undistorted images using Wang, Qiu, and Shao (2009)'s method. (a,b) Distorted images. (c,d) Undistorted images using the single parameter division model based that is based on circle fitting. Reprinted from Wang et al. (2009).	24
2.25	Comparison of distorted and undistorted points. (a) Synthetic image. (b) Distorted synthetic image. Blue dots represent distorted points. (c) Undistorted image. Blue dots represent undistorted points. (d) Comparison of distorted and undistorted lines. Red dots represent distorted points and green dots represent corresponding undistorted points. Reprinted from Alvarez et al. (2009).	25
2.26	Undistorted images at different focal lengths. (a) Distorted image at focal length 45.16 mm. (b) Distorted image at focal length 156.55 mm. (c) Undistorted image at focal length 45.16 mm. (d) Undistorted image at focal length 156.55 mm. The zoom focus is unknown and is estimated. Reprinted from León, Déniz, and Castellano (2011).	26
2.27	Least squares vs. RANSAC model fitting including two outliers. (a) Result of least squares based line fitting. (b) Result of RANSAC based line fitting. Reprinted from Hartley and Zisserman (2004).	29
2.28	Flow chart of the RANSAC algorithm. Reprinted from (Cyganek & Siebert, 2009)	30
4.1	Example experiment with synthetic image size 640×480 . (a) Original image. (b) Distorted image with $\lambda = -10^{-6}$ and $(x_0, y_0) = (320, 240)$ (the image center). c) Canny edges (d) Extracted contours. (e) Estimated arcs. (f) Undistorted image using estimated values of $\lambda = -1.00419e^{-6}$, $x_0 = 319.352$, and $y_0 = 238.009$. Using true parameters RMSE = 3.27813 and PNSR = 37.8183 dB; Using estimated parameters RMSE = 3.86511 and PNSR = 36.3876 dB.	49
4.2	Experiment 1. Undistortion of synthetic images. Image size is 640×480 and distortion center is (320, 240). First row and third row: distorted images at different levels of λ . Second row and fourth row: corresponding undistorted images using parameters estimated by the “Ransac-LM” circle fitting method.	50
4.3	Experiment 1. Undistortion of synthetic images. Image size is 640×480 and distortion center is (320, 240). First row and third row: distorted images at different levels of λ . Second row and fourth row: corresponding undistorted images using parameters estimated by the “Ransac-LM” circle fitting method.	51

4.4	Experiment 2. Undistortion of synthetic images. Image size is 640×480 and distortion center is (320, 240), (330, 250), (340, 260), and (350, 270) respectively . First row and third row: distorted images at different levels of distortion center (DC). Second row and fourth row: corresponding undistorted images using parameters estimated by the “Ransac-LM” circle fitting method.	52
4.5	Experiment 2. Undistortion of synthetic images. Image size is 640×480 and distortion center is (360, 280), (370, 290), (380, 300), and (390, 310) respectively. First row and third row: distorted images at different levels of distortion center (DC). Second row and fourth row: corresponding undistorted images using parameters estimated by the “Ransac-LM” circle fitting method.	53
4.6	Results of experiment 1 using synthetic images. (a-c) represent results with varying λ , with distortion center fixed at the image center. All graphs show averages over 10 runs with error bars showing 95% confidence intervals on the mean. (a) True versus estimated λ . (b) Average RMSE. (c) Average PNSR.	54
4.7	Results of experiment 2 using synthetic images. (a-c) represent results with varying distance of the distortion center to the center of the image. λ is fixed at -10^{-6} . All graphs show averages over 10 runs with error bars showing 95% confidence intervals on the mean. (a) True versus estimated distance of the distortion center from the image center. (b) Average RMSE. (c) Average PNSR.	55
4.8	Example results on real image. (a) Original image. (b) Detected edges. (c) Extracted contours. (d) Identified arcs. (e) Undistorted image using parameters estimated via the “Ransac-LM” circle fitting method. The distorted image is taken from Ociepka Ociepka (2003).	56
4.9	Undistortion of real images. Column 1, column 3, and column 5 represent original images. Column 2, column 4, and column 6 represent undistorted images using parameters estimated via the “Ransac-LM” circle fitting method. The distorted images are taken from several sources. Barreto and Daniilidis (2005); Rosten and Loveland (2011); Swaminathan and Nayar (2000); Sarge (n.d.); Thormählen et al. (2003); Chen, Huang, Shiao, and Chen (2009); Bockaert (n.d.); El-Melegy and Farag (2003); Chanel (n.d.); Alvarez, Gomez, and Sendra (2010); Whittaker (n.d.); Skewes (n.d.); Solheim (n.d.); Tomasi (2007); Laksi (n.d.); Yusuf (n.d.); Bucket (n.d.); Dyer (n.d.); Oleson (n.d.); Kbh3rd (2008); Rideout (2002).	57
4.9	Failure case analysis: (a) Original image. (b) Undistorted image by setting the MIN_CONTOUR_LENGTH to 25 pixels. (c) Undistorted image by setting the MIN_CONTOUR_LENGTH to 30 pixels.	66

List of Tables

Table	Title	Page
2.1	Comparison of different circle fittings methods. LM = Levenberg Marquardt, KMvH = Kukush Markovsky van Huffel; IURAT = Invariant Under Rotations And Translations. $r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$, $\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i$; $A = \pm \frac{1}{2R}$, $B = -2Aa$, $C = -2Ab$, $D = \frac{B^2+C^2-1}{4A}$; $\bar{z} = \frac{1}{N} \sum_{i=1}^N (x_i^2 + y_i^2)$. (x_c, y_c) is the center and R is the radius of the circle. \mathbf{A} is a vector representing the parameters of a circle. Initial guess = IG, Outliers resistant = OR, Geometric = GC, Analytic = AC, IURAT = IT.	32
2.1	Comparison of different circle fittings methods. LM = Levenberg Marquardt, KMvH = Kukush Markovsky van Huffel; IURAT = Invariant Under Rotations And Translations. $r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$, $\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i$; $A = \pm \frac{1}{2R}$, $B = -2Aa$, $C = -2Ab$, $D = \frac{B^2+C^2-1}{4A}$; $\bar{z} = \frac{1}{N} \sum_{i=1}^N (x_i^2 + y_i^2)$. (x_c, y_c) is the center and R is the radius of the circle. \mathbf{A} is a vector representing the parameters of a circle. Initial guess = IG, Outliers resistant = OR, Geometric = GC, Analytic = AC, IURAT = IT (continued).	33
2.1	Comparison of different circle fittings methods. LM = Levenberg Marquardt, KMvH = Kukush Markovsky van Huffel; IURAT = Invariant Under Rotations And Translations. $r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$, $\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i$; $A = \pm \frac{1}{2R}$, $B = -2Aa$, $C = -2Ab$, $D = \frac{B^2+C^2-1}{4A}$; $\bar{z} = \frac{1}{N} \sum_{i=1}^N (x_i^2 + y_i^2)$. (x_c, y_c) is the center and R is the radius of the circle. \mathbf{A} is a vector representing the parameters of a circle. Initial guess = IG, Outliers resistant = OR, Geometric = GC, Analytic = AC, IURAT = IT (continued).	34
4.1	Comparison with Alvarez et al. Alvarez et al. (2009). Image size is 640×480 . Distortion center = DC, RMSE (pixel intensity)= RMSE, Cpu time = Time, and Total number of arcs = Arcs.	66

Chapter 1

Introduction

This dissertation explores automatic radial distortion estimation based on the plumbline approach. In this chapter, I lay out the motivation for the research, our contributions, and I provide an outline of the dissertation.

1.1 Motivation

Most computer vision algorithms, particularly structure from motion algorithms, rely critically on the assumption of the linear pinhole camera model. However, most commercially available cameras introduce sufficiently severe optical distortion that the pinhole assumption is invalid, making distortion correction a must. Radial distortion is the most significant type of distortion in today's cameras (Zhang, 2000; Strand & Hayman, 2005; Kukelova & Pajdla, 2011). It is most evident in images produced with low-cost, wide-angle lenses (Gonzalez-Aguilera, Gomez-Lahoz, & Rodriguez-Gonzalvez, 2011). Such lenses are being widely deployed, for example, in automotive driver assistance applications (Friel, Hughes, Denny, Jones, & Glavin, 2010; Hughes, Glavin, Jones, & Denny, 2009). But radial distortion is also significant enough in higher-quality cameras to introduce error into 3D reconstruction processes. Radial distortion bends straight lines into circular arcs (Strand & Hayman, 2005; Wang et al., 2009), violating the main invariance preserved in the pinhole camera model, that straight lines in the world map to straight lines in the image plane (Devernay & Faugeras, 2001; Hartley & Zisserman, 2004).

In this dissertation, in order to eliminate radial distortion from images, we have devised a fully automatic method able to remove radial distortion from an image. The method involves new algorithms for robust estimation of circular arcs and robust estimation of distortion parameters. In order to evaluate the new method, we perform a comprehensive quantitative study of its performance on distorted synthetic images and provide extensive examples of its ability to remove distortion from a large, challenging set of real images taken from Web sites and previous papers on distortion estimation. We find that the algorithm performs very well, with excellent reconstruction of the original synthetic images even under severe barrel distortion and pincushion distortion. We also find that the method is able to eliminate nearly all of the visible distortion in the real images, including those acquired with wide angle and fish-eye lenses. Finally, we perform a direct comparison of our method with that of Alvarez et al. (Alvarez, Gomez, & Sendra, 2009), the only researchers who have provided a publicly accessible implementation of their method, on synthetic images. The Alvarez et al. method exploits user assistance in identifying points on straight lines, but we nevertheless find that our fully automatic method provides superior reconstruction of the original undistorted image. Our method is thus a practical solution to the important problem of radial distortion estimation and removal.

1.2 Contributions

This dissertation makes the following contributions towards construction of an automatic system for radial distortion estimation and removal from a single image.

1. We propose a new method for automatic radial distortion estimation based on the plumbline approach.
2. Our method works from a single image without requiring a special calibration pattern.
3. We introduce a new approach for robust automatic estimation of circular arcs.
4. We devise a new method for robust and optimal estimation of distortion parameters based on identified circular arcs.
5. We perform an extensive empirical study of our method on synthetically distorted images. The synthetic distortion is based on different values of the radial distortion parameters.
6. We perform an extensive study of different circle fitting methods.
7. We include a comparative statistical analysis of how different circle fitting methods contribute to accurate distortion parameter estimation.
8. We provide qualitative results on a wide variety of challenging real images from a large, challenging set of real images taken from Web sites and previous papers on distortion estimation.
9. We contribute source code based on OpenCV (Bradski, 2000) online¹ for researchers interested in evaluating or extending our procedure.

1.2.1 List of Publications

In this section, I provide a list of papers published as part of this dissertation and the list of papers currently under preparation for submission.

Published Work

- **Automatic Radial Distortion Estimation from a Single Image**
Faisal Bukhari and Matthew N. Dailey
In Journal of Mathematical Imaging and Vision (JMIV), doi: 10.1007/s10851-012-0342-2, pp. 1–15, 2012
Included in Chapters 3 and 4

¹See <http://www.cs.ait.ac.th/vgl/faisal/downloads.html>.

- **Robust Radial Distortion from a Single Image**
Faisal Bukhari and Matthew N. Dailey
In Proceedings of the 6th International Conference on Advances in Visual Computing (ISVC'10) - Volume Part II, pp. 11–20, 2010
Included in Chapters 3 and 4

Manuscripts Currently in Preparation

- **Automatic Plumbline Based Radial Distortion Estimation and Removal**
Faisal Bukhari and Matthew N. Dailey
In Image Processing On Line (IPOL) repository
- **MonoSLAM Based Navigation of Low Cost Quadrotor in Indoor Scenes**
Faisal Bukhari and Matthew N. Dailey
To be submitted in International Conference on Advanced Robotics (ICAR)

Manuscripts in Planning

- **Autonomous Navigation of Quadrotor in Indoor Scenes for Rapid Feedback to Security Agencies**
Faisal Bukhari and Matthew N. Dailey
To be submitted in Journal of Field Robotics

1.3 Dissertation Outline

In this section, I provide an outline of the rest of the dissertation.

In Chapter 2, I review the literature related to radial distortion estimation, RANSAC, and different circle fitting methods.

In Chapter 3, I describe our methodology, mathematical model, algorithms for circular arc estimation, our procedure for refining circular arcs, and our algorithm for robust distortion parameter estimation.

In Chapter 4, I describe our experiments on synthetic and real images, our results, a detailed statistical analysis, and a comparison of our method with Alvarez et al. (Alvarez, Gomez, & Sendra, 2009).

In Chapter 5, I conclude the dissertation, summarize the contributions, and outline future work.

Chapter 2

Literature Review

Most computer vision algorithms, particularly structure from motion algorithms, rely critically on the assumption of a linear pinhole camera model. However, most commercially available cameras introduce sufficiently severe optical distortion that the pinhole assumption is invalid, making distortion correction a must. Radial distortion is the most significant type of distortion in today's cameras (Zhang, 2000; Strand & Hayman, 2005; Kukelova & Pajdla, 2011). It is most evident in images produced with low-cost, wide-angle lenses (Gonzalez-Aguilera et al., 2011). Such lenses are being widely deployed, for example, in automotive driver assistance applications (Friel et al., 2010; Hughes et al., 2009). But radial distortion is also significant enough in higher-quality cameras to introduce error into 3D reconstruction processes. Radial distortion bends straight lines into circular arcs (Strand & Hayman, 2005; Wang et al., 2009), violating the main invariance preserved in the pinhole camera model, that straight lines in the world map to straight lines in the image plane (Devernay & Faugeras, 2001; Hartley & Zisserman, 2004). Radial distortion may appear as barrel distortion, usually arising at short focal lengths, or pincushion distortion, usually arising at longer focal lengths. Besides radial distortion, another type of distortion is tangential distortion. We do not have experience with real cameras that introduce significant tangential distortion, so like most previous work (Zhang, 2000; Thormählen et al., 2003; Strand & Hayman, 2005; Alvarez et al., 2009; Kukelova & Pajdla, 2011; Rosten & Loveland, 2011), we ignore tangential distortion.

This chapter discusses some of the important literature relevant to radial distortion. There are several methods for removing radial distortion from images. Methods for radial distortion estimation fall into three major categories: point correspondence, multiple view autocalibration, and plumb-line. I also discuss RANSAC as well as algebraic and geometric circle fitting methods, which will be used in my radial distortion estimation algorithm.

2.1 Point correspondence based methods

Point correspondence based methods (Tsai, 1987; Zhang, 2000; Braüer-Burchardt, 2004) are ideal for distortion estimation during pre-calibration of a camera with a fixed focal length. They identify image points with known 3D positions in multiple images using a known pattern such as a chessboard and then estimate the parameters of an undistortion function. The parameterized undistortion function can then be used to undistort specific images or point positions. These point correspondence methods are highly reliable and accurate; radial distortion estimation and removal is a solved problem for cameras that are pre-calibrated at a fixed focal length. Manual camera calibration, however, is a tedious process that is not always possible, for example, when we want to process an existing image sequence acquired with an unknown camera, when we want to change the focal length dynamically during image sequence acquisition, or when we want to get a mobile robot experiment up and running quickly.

For internal camera calibration either a 2D or 3D calibration object such as the ones shown in Figure 2.1 are used. Moons et al. (2009) point out that a typical internal calibration

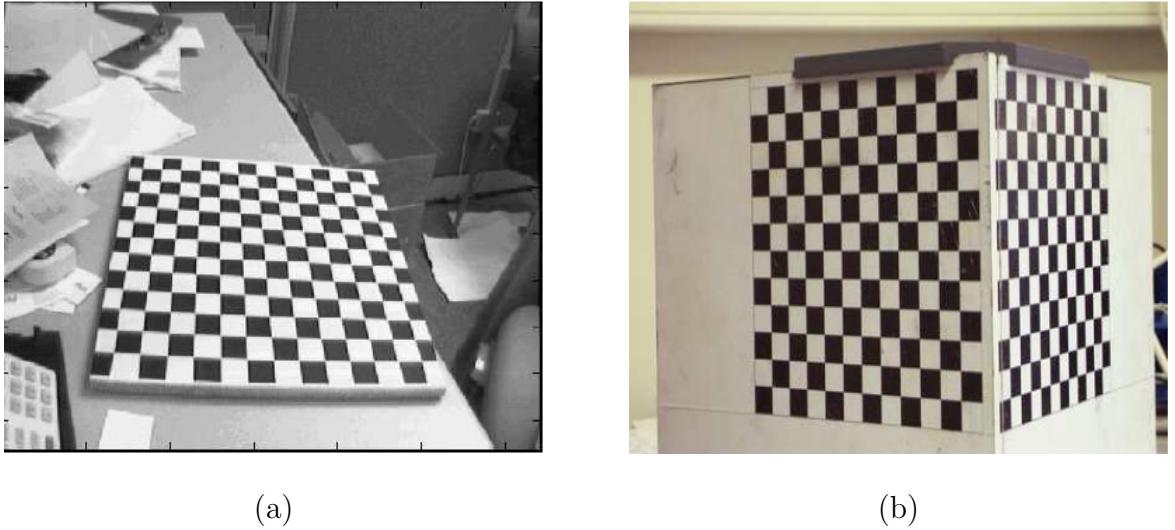


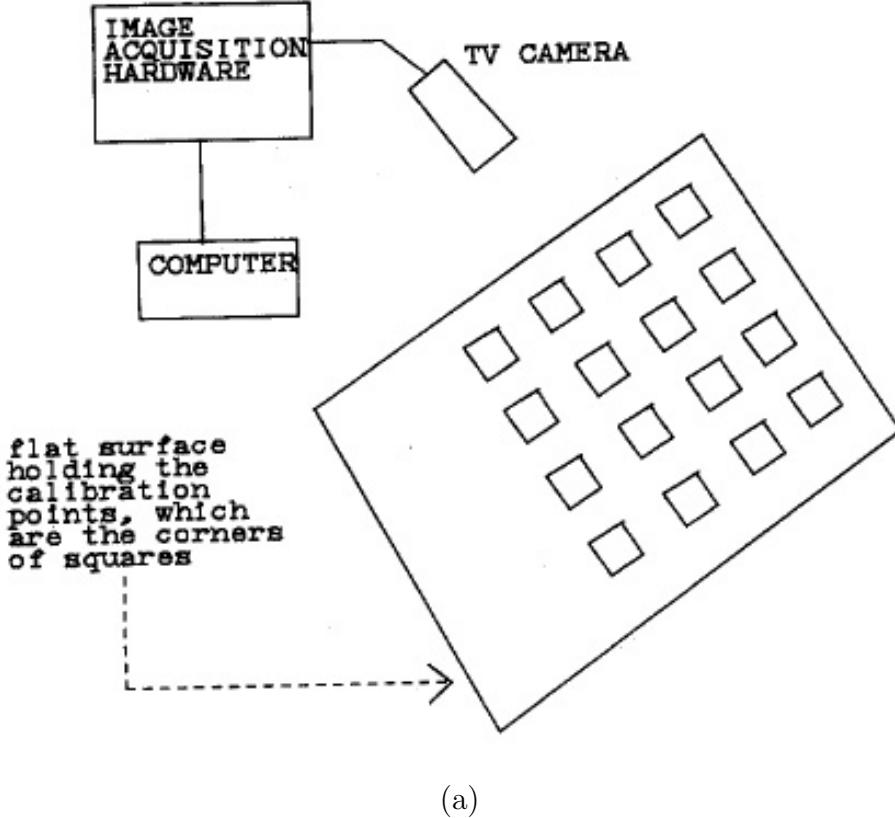
Figure 2.1: Two calibration patterns. (a) Calibration pattern in 2D. (b) Calibration pattern in 3D. Reprinted from Moons et al. (2009).

algorithm has following steps:

- Create a calibration image either 2D as in Figure 2.1 (a) or 3D as in Figure 2.1 (b).
- Measure the 3D positions of markers on the calibration image.
- Grab images (one or two) of the object based on the algorithm, with the camera to be calibrated.
- Extract and identify markers in the calibration image.
- From the 3D-2D correspondences obtained in the previous step, fit a linearized calibration model.
- Refine calibration using non-linear optimization.

Tsai (1987) has developed a point correspondence based method using an off-the-shelf TV camera and lens. His method relies on corners of uniformly spaced boxes of known dimensions for calibration. His method comprises a two-stage algorithm. The two-stage algorithm consists of calculation of the camera's external pose associated to the object reference frame. Also it requires computation of image scanning parameters, focal length, and radial distortion. In this category of methods, usually a calibration object corresponds to two or three planes perpendicular to each other. But in this paper a plane experiencing a specifically known translation is used. Figure 2.2 shows the layout procedure consisting of a monocular view of a coplanar set of points. Initially, Tsai (1987) did not include distortion center in the list of parameters to be calibrated, but after the submission of this paper, he derived some new formulae (Lenz & Tsai, 1988) that indicate how the distortion center error can affect the accuracy of 3D measurement. This technique is high-priced in terms of calibration equipment and setup.

Zhang (2000) presents a novel technique for calibrating a camera. It only requires a planar surface with at least two different orientations. Either camera or calibration pattern can be

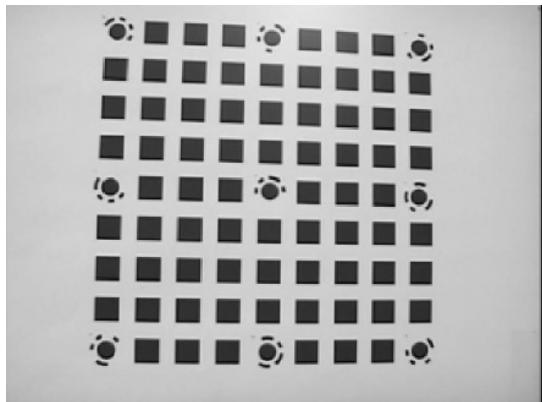


(a)

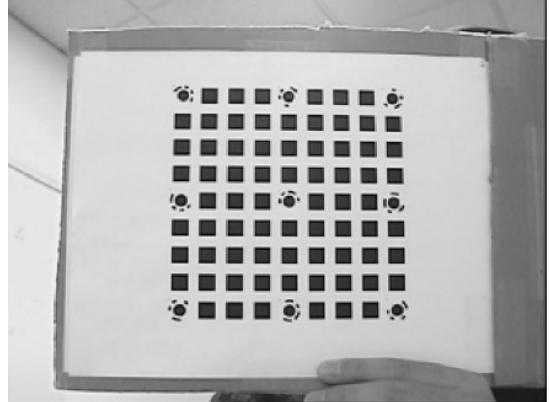
Figure 2.2: Schematic diagram for camera calibration. Reprinted from Tsai (1987).

moved without any restrictions. There is no requirement that the motion to be known. The method is based on a closed-form solution, succeed by a nonlinear fine-tuning and depends on maximum likelihood. Figure 2.3 shows images of a calibration pattern at different distances. The image resolution is 640×480 . The plane contains 9×9 squares with 9 special points. These special points are used to automatically detect the correspondence between the points and corners in the image. Bouguet (2008) provides an on-line demo based on Zhang (2000)'s method.

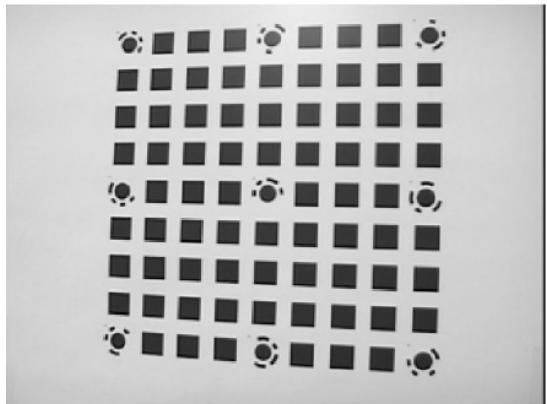
Braüer-Burchardt (2004) describes a method that requires some reference points with known coordinates. Figure 2.4 (a) and (b) show the two calibration patterns he uses. In the first calibration pattern (Figure 2.4 (a)), the intersection of horizontal and vertical lines are used. In the second calibration pattern (Figure 2.4 (b)), the center points of dots are used. First radial distortion is removed, then the undistorted image is compared with the original image obtained by ideal pinhole mapping. This computation is accomplished by fitting a projective 2D-2D transform to the known original point coordinates to the undistorted image coordinates. These original point coordinates are 3D points but usually $|Z|$ -coordinate is 0. The remaining error, if any, is removed by a fitting projective transformation.



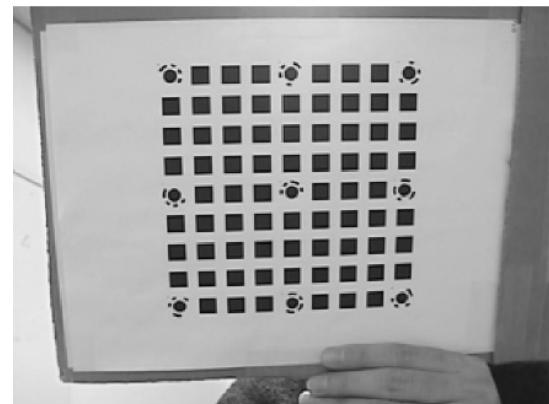
(a)



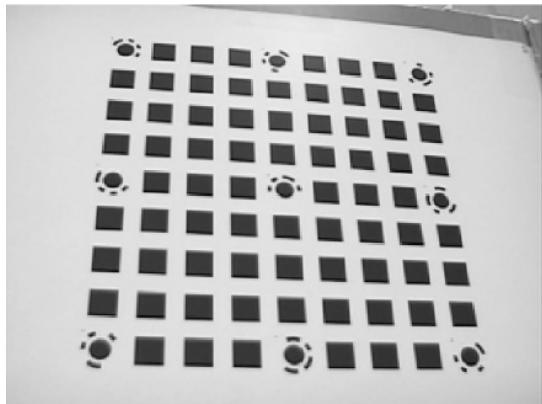
(b)



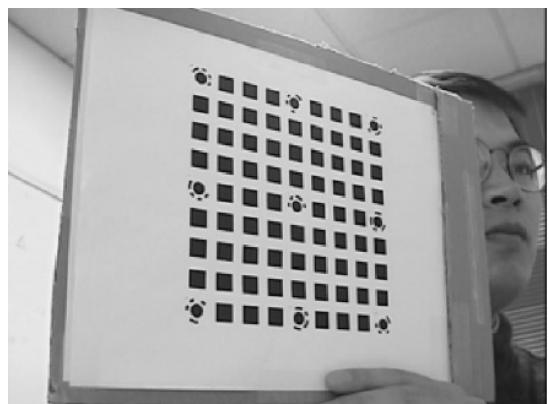
(c)



(d)

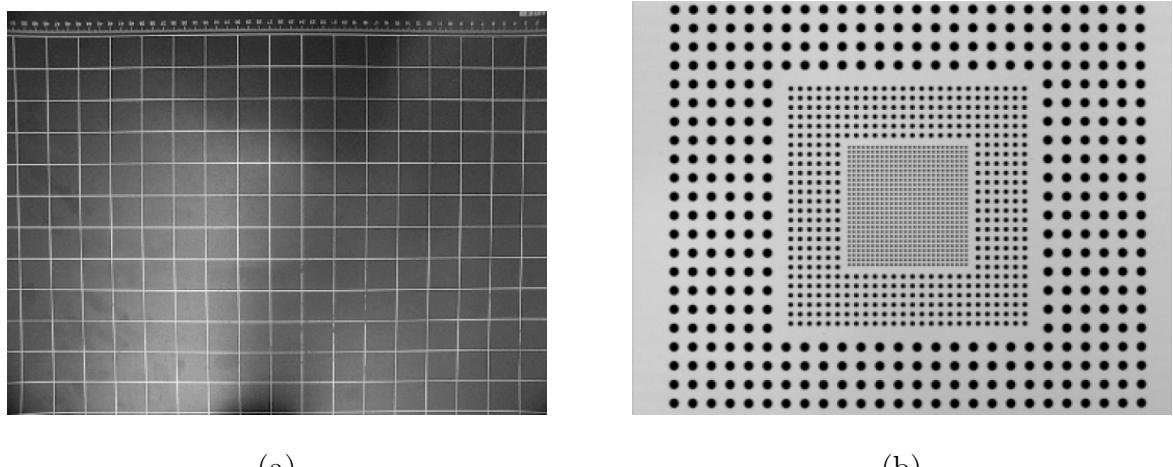


(e)



(f)

Figure 2.3: Two sets of images of a calibration pattern. Images (a), (c), and (e) are taken from closer distances. Images (b), (d), and (f) are taken from further distances. Reprinted from Zhang (2000).



(a)

(b)

Figure 2.4: Two calibration patterns. (a) Grid pattern. (b) Point pattern. Reprinted from Braüer-Burchardt (2004).

2.2 Multiple view auto-calibration based methods

Multiple view auto-calibration is an active area of computer vision research that aims to extract camera parameters automatically from natural images. Auto-calibration methods use a sequence of arbitrary natural images without any special pattern or information about the scene. Although many auto-calibration methods assume a pinhole camera, others do attempt to simultaneously estimate radial distortion parameters and pinhole parameters (Faugeras et al., 1992; Stein, 1996; Fitzgibbon, 2001; Hartley & Kang, 2007; Ramalingam et al., 2010; Kukelova & Pajdla, 2011).

Auto-calibration is a mature area of research, but the main limitation of this class of methods is that it requires multiple images under camera motion. For fixed cameras and for situations where immediate on line estimation is desirable, multiple view methods are inappropriate.

Faugeras et al. (1992) introduce a new method that does not require a calibration object with known 3D position. It requires point correspondences in a sequence of images. The authors find that it is possible to calibrate using a noisy data by pointing a camera at a scene, choosing interesting points, and then tracking them in the image as the camera moves. It is not necessary to know the camera motion. Their method consist of two steps. The first step consists of epipolar transformation, and the second step consists of calculation of Kruppa equations, which link the epipolar transformation to the image of the absolute conic. Figure 2.5 shows precise construction of a dual curve g . Algorithm 1 explains authors' method for solving the Kruppa equations, called the continuation method.

Stein (1996) develops a new method that only relies on point correspondences in multiple views. It does not require either camera locations or 3D locations of points. It is an iterative method that uses a feature based 3D reconstruction. The method utilizes epipolar and trilinear constraints to formalize two separate cost functions respectively. Figure 2.6 shows Stein (1996)'s calibration images. Figure 2.6 (d) image is taken with longer focal length. He says his method has two types. The first method uses two images and the second method uses three images. Following is a brief explanation of the two methods:

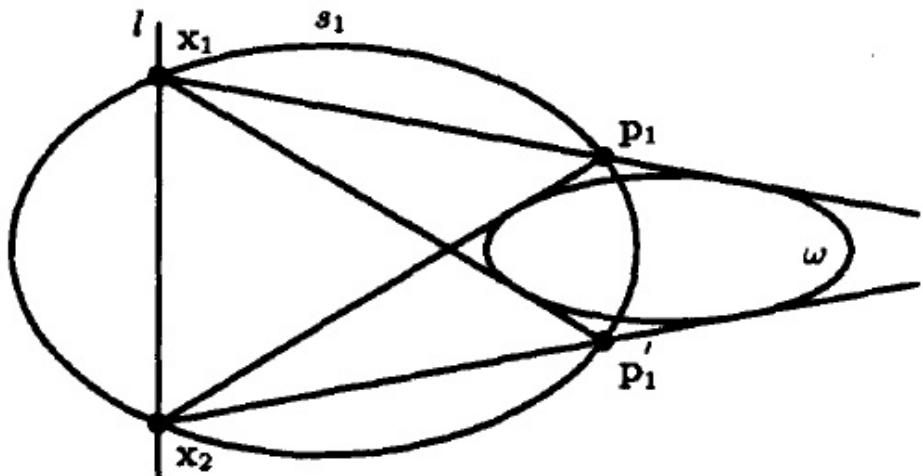
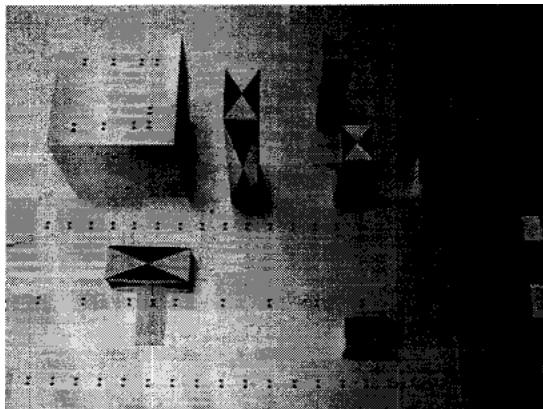
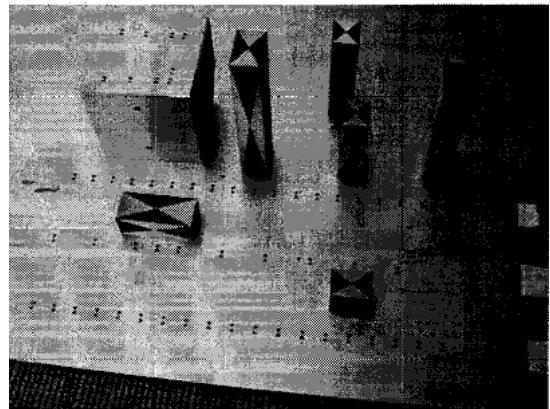


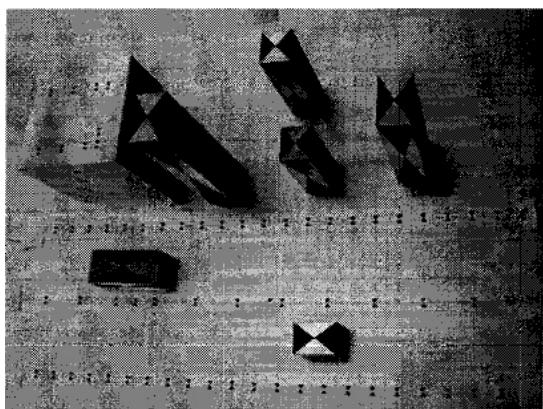
Figure 2.5: Development of dual curve. Suppose p_1 and p'_1 denote the two epipoles for the first motion of the camera. The epipolar transformation is based on the Steiner conic s_1 passing through p_1 and p'_1 . If y is a point of s_1 , then two epipolar lines $\langle p_1, y \rangle$ and $\langle p'_1, y \rangle$ correspond to each other. The two tangents from p_1 to w intersect s_1 at points x_1, x_2 . The line segment $\langle x_1, x_2 \rangle$ of s_1 corresponds to the point $x_1 \times x_2$ in the dual of the image plane. The point $x_1 \times x_2$ lies on a curve g that is an algebraic transformation of c . To find the camera calibration uniquely, three camera displacements which give six conditions are enough. Note that Steiner theorem states that there is always a distinct conic s_1 passing through p_1 and p'_1 . y is a point of s_1 , if and only if, $\langle p_1, y \rangle \perp \langle p'_1, y \rangle$. Reprinted from Faugeras et al. (1992).



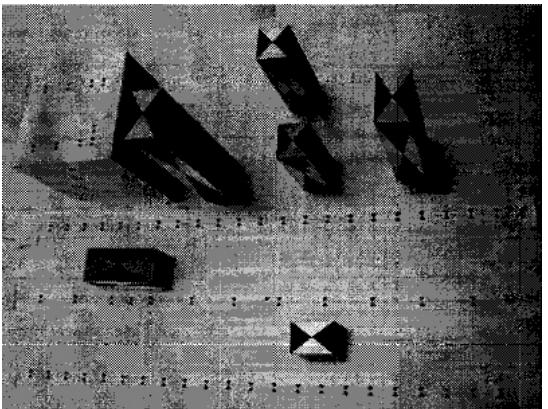
(a)



(b)



(c)



(d)

Figure 2.6: Calibration images. Images (a), (b), and (c) are used for calibration. Image (d) is taken with a longer focal length, and lines are curved. Reprinted from Stein (1996).

Input: Three displacements D_i

Output: Solution of the Kruppa equation

foreach D_i **do**

 Calculate corresponding points between the two images

 Calculate the epipoles

 Calculate the homography of epipolar lines

 Calculate the two Kruppa equations

 Calculate the six Kruppa equations using the continuation method

 Calculate the intrinsic parameters

end

Algorithm 1: Robust arc identification.

- The two-image method:

- Find epipole and epipolar lines using at least 8 point correspondences in two images.
- Find a cost function for more than 8 points that is based on RMS (Root Mean Square) of distance between the points and corresponding points epipolar lines.
- Find distortion parameters by minimization this cost function.

- The three-image method:

- Relate 4 independent trilinear equations using corresponding points in three images.
- Get 27 parameters in a linear sense based on these equation and on at least 7 point correspondences.
- Calculate reprojection of these points from the two images into the third image based on the above parameters.
- Get the cost function based on the reprojection error.
- Get the distortion parameters by minimizing the cost function.

Fitzgibbon (2001) develops a radial distortion estimation which is based on a linear estimation of the fundamental matrix based on correspondences' between stereo images. He introduces the division model. He uses his single parameter division model in this estimation. Figures 2.6 (a) and (b) show a stereo pair and Figure 2.6 (c) shows corresponding points. A linear estimate of the fundamental matrix is achieved by using:

- A single parameter division model as compared to the commonly used polynomial model for radial distortion estimation is used that takes simpler form in homogeneous coordinates.

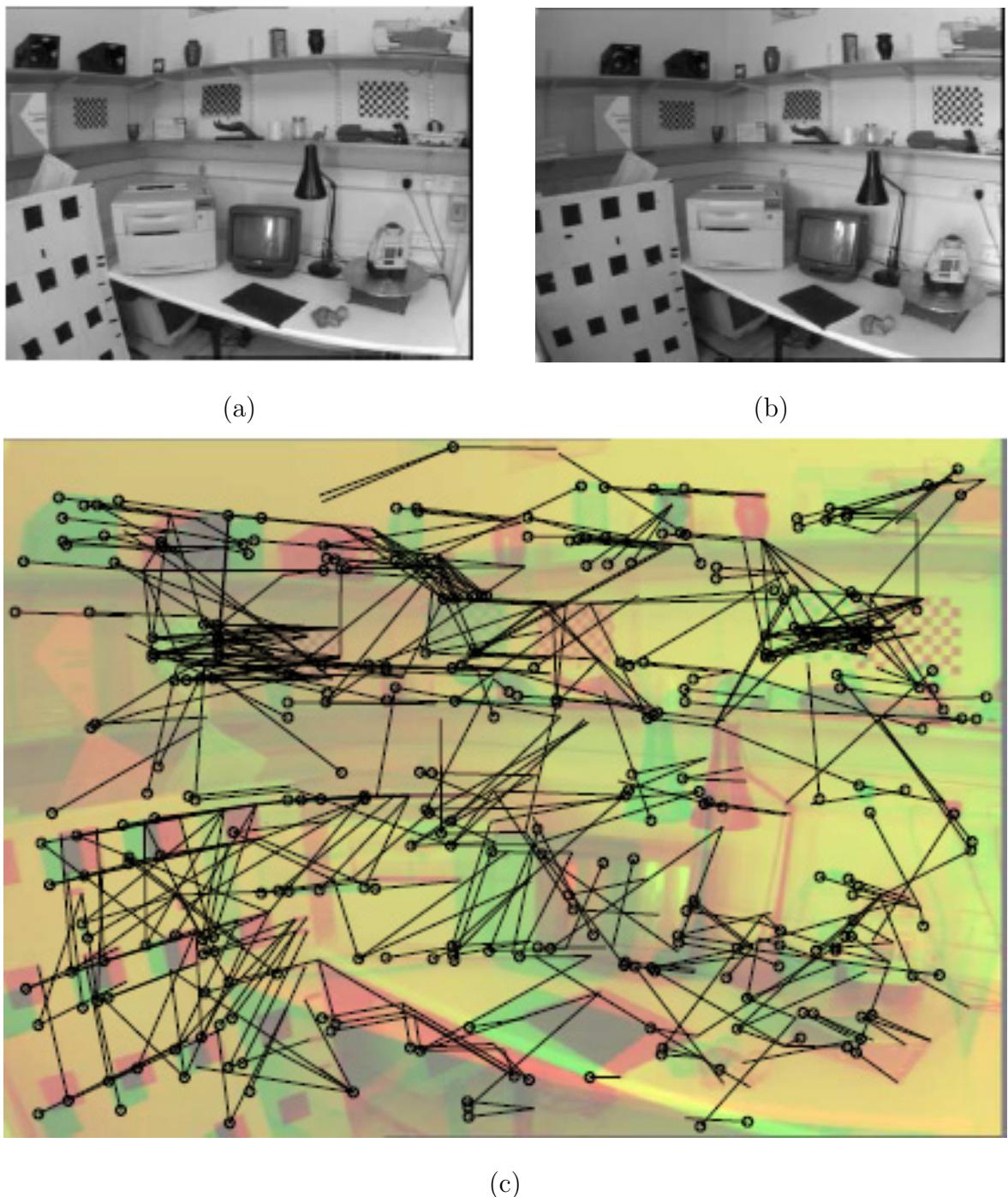


Figure 2.7: Stereo images. Images (a) and (b) represent a stereo pair. (c) Circles represent points in one image, and lines represent their tentative corresponding points in the other image. RANSAC is used for input matches. The different color channels are superimposed in the images. Reprinted from Fitzgibbon (2001).

- The fundamental matrix estimation as a Quadratic Eigenvalue Problem (QEP) is used in order to utilize available efficient algorithm.

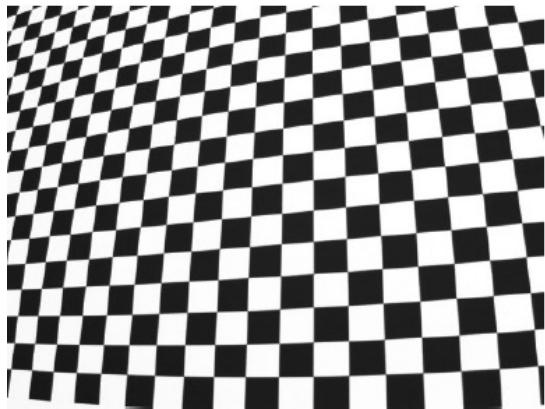
Hartley and Kang (2007) propose a method that relies on a planar or a non planar calibration grid, which is captured from several images. Figure 2.8 (a) and Figure 2.8 (b) show their checkerboard and detected grid, respectively. Figures 2.9 (a), (b), (c), (d) show non-planar calibration grid. Their method is non-iterative and parameter free that does not rely on any particular radial distortion model like the polynomial model or the division model. Their method computes a center of distortion for computing optimal results. Also they prove that sometimes in real cameras, the distortion center is significantly displaced from the image center. Figure 2.8 (c) and Figure 2.8 (d) show distorted images, and Figure 2.8 (e) and Figure 2.8 (f) show corresponding undistorted images.

Ramalingam et al. (2010) develop a method for self calibration that considers specific camera motions. For example pure translations and rotations are considered. In this case, the computation of rotation and translation parameters are not required. Combining the types of motion with image matches gives geometric constraints on the projective rays. The authors show that using only translation motion, self-calibration can be performed, but up to an affine transformation of the projective rays. Figures 2.10 (a), (b), and (c) show extraction of rotations and translations.

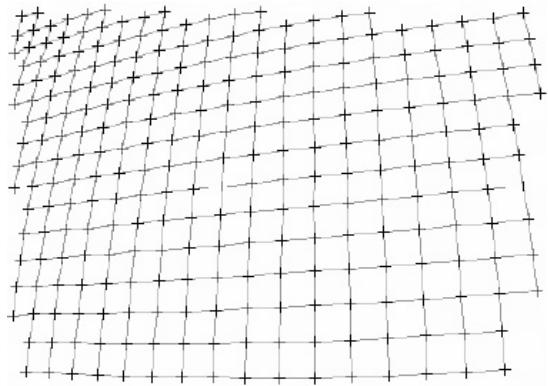
Kukelova and Pajdla (2011) develop a new method that is based on a single parameter division model. They perform simultaneous estimation of radial distortion, epipolar geometry, and camera pose. The estimation procedure is formulated as a minimization problem and can be solved based on a minimal set of image points. They provide two solutions for estimating radial distortion and epipolar geometry based on 8 point correspondences between the two images. They enforce the determinant of the fundamental matrix to be 0, which is the basis of a system of 9 quadratic and one cubic equation in nine variables. The method reduces the number of samples in the RANSAC loop as compared to the 9-point algorithm. The system is simplified by first eliminating 6 variables, then the system is solved using two different techniques:

- Gröbner basis method:
 - It solves smaller eigenvalue problem. It gives as less solution, 16, as compare to 29 previously. Also, it is faster and more stable.
- The polynomial eigenvalue method:
 - The second method is based on the calculation of polynomial eigenvalue. It is easy to understand and implement. It is more stable as compared to Gröbner basis method. Also, it is faster than the Gröbner basis method.

Figures 2.11 (a-d) represent distorted images and Figure 2.11 (e-h) represent corresponding undistorted images.



(a)



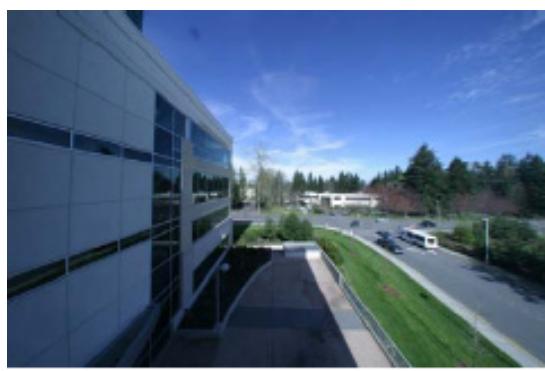
(b)



(c)



(d)



(e)



(f)

Figure 2.8: Hartley and Kang (2007)'s method sample results. (a) Checker-board. (b) Detected grid. (c, d) Distorted images. (e, f) Corresponding undistorted images. Reprinted from Hartley and Kang (2007).

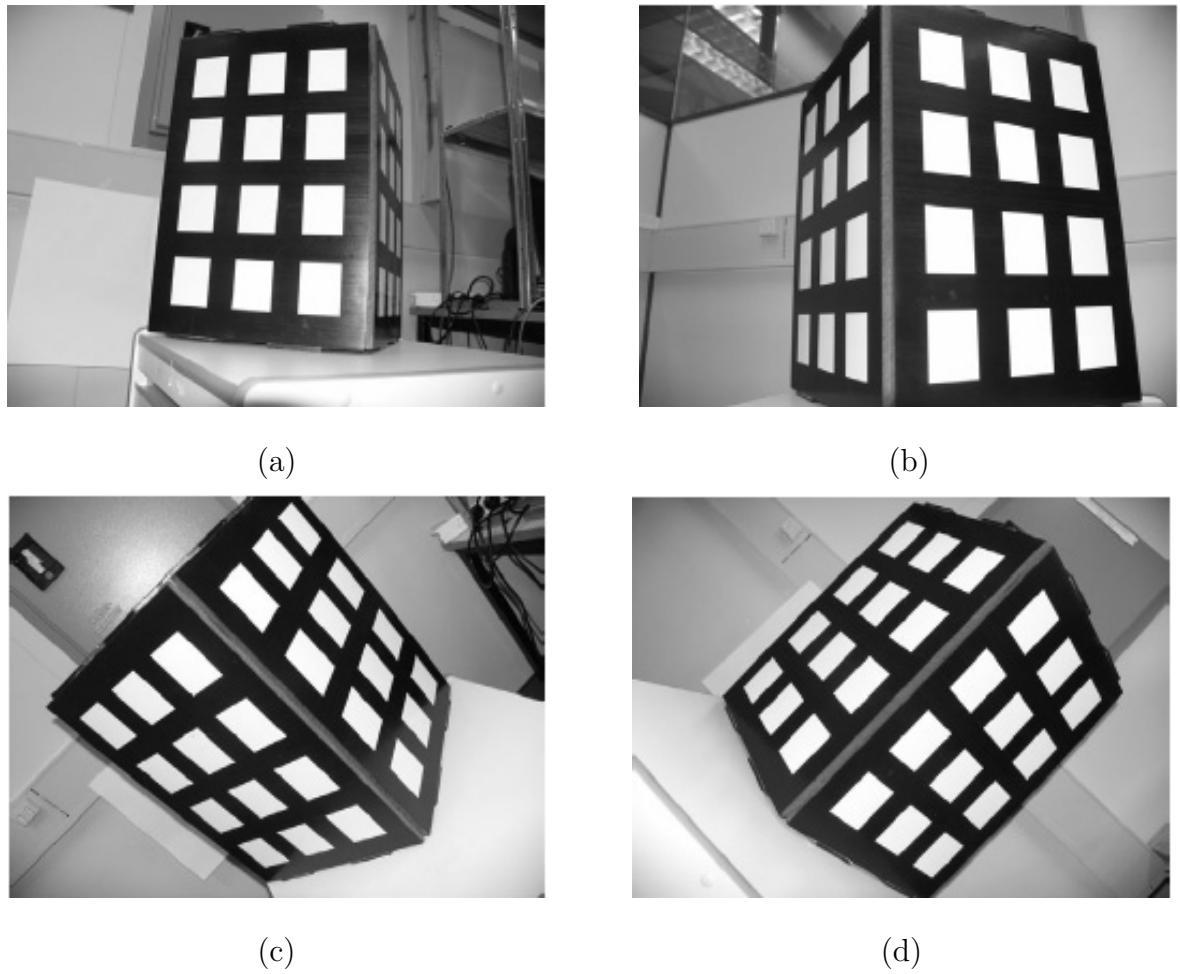


Figure 2.9: Non-planar calibration grid. Reprinted from Hartley and Kang (2005).

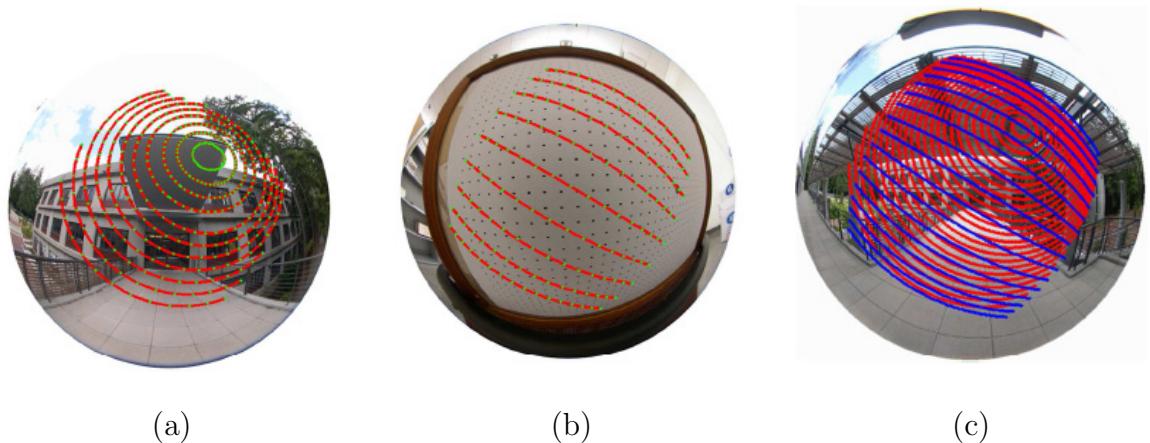


Figure 2.10: Flow curves. (a) Pure rotations (b) Pure translation. (c) Intersection of rotation and translation flow curves. Reprinted from Ramalingam et al. (2010).

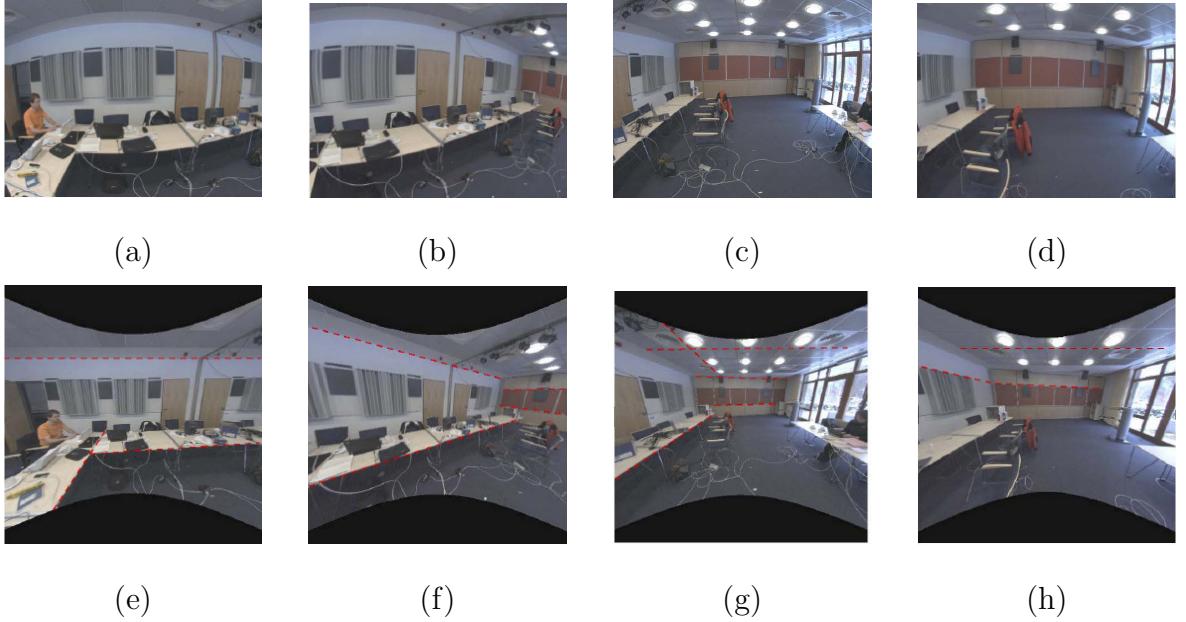


Figure 2.11: Example of image sequence. (a-d) Distorted images . (e-h) Undistorted images. Reprinted from Kukelova and Pajdla (2011).

2.3 Plumb-line methods

In view of the limitations of the point correspondence and auto-calibration methods, robust automatic distortion estimation and removal from a single natural image would be extremely useful for many applications, particular those in human-made environments containing abundant lines. For example, it could be used in place of an extensive calibration procedure to get a mobile robot or quadrotor experiment up and running quickly in an indoor environment. Plumb-line methods are the most promising for robust distortion estimation from a single image or a small number of images. Rather than using a known pattern or sequence of images under camera motion, they estimate distortion parameters directly from distorted straight lines in one or more images. Straight lines are frequent enough in most human-made environments to make distortion estimation from a single image possible (Brown, 1971; Kang, 1997, 2000; Devernay & Faugeras, 2001; Thormählen et al., 2003; El-Melegy & Farag, 2003; Strand & Hayman, 2005; Wang et al., 2009; Alvarez et al., 2009; León et al., 2011).

Brown (1971) develops the theory for the plumb-line approach. He also develops a method to remove radial and decentering distortion using a close-range camera. His paper is classic. Figure 2.12 shows an image of a close-range camera. He uses parallel plumb lines to estimate radial distortion coefficients. His method is iterative and based on a gradient descent algorithm. The required manual selection of points on plumb lines is time consuming.

Kang (1997) presents a radial distortion removal method based on a single image. He develops three interactive methods, in which with user interaction is used to select lines from an image. In the first method, users manually select points on a distorted line. The second and the third methods are based on drawing snakes on the image, which are approximated lines in an image. The snakes are drawn on strong edges. The demarcation line among the three methods methods is the action of snakes. Figure 2.13, Figure 2.14, and Figure 2.18 show

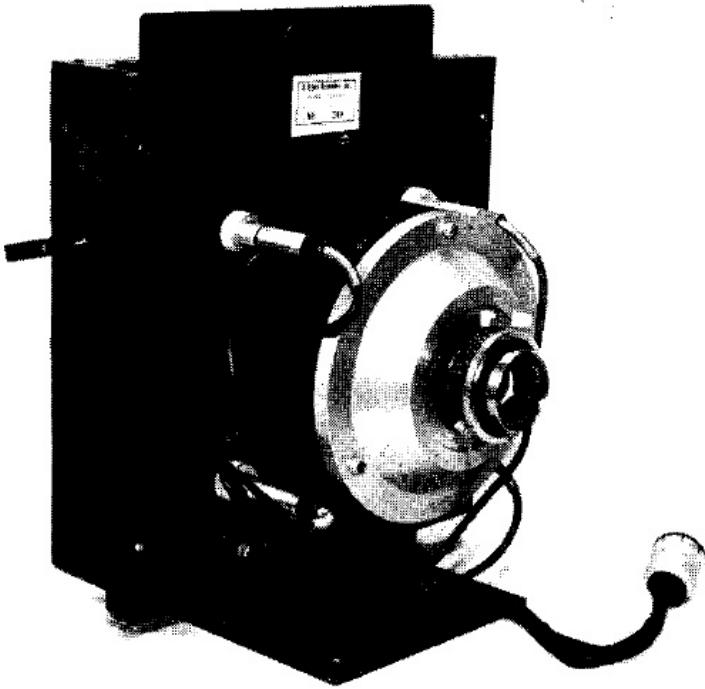


Figure 2.12: A close-range camera. The camera accepts 190×215 mm by $\frac{1}{4}$ inch ballistic plates. Also it exposes a 6-inch diameter format. Focusing can be done from 2 feet to infinity using interchangeable cones. Reprinted from Brown (1971).

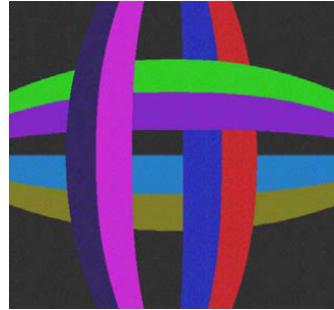
application of conventional and radial distortion snake algorithms in synthetic images with no noise, synthetic images with Gaussian noise, and real images respectively.

Kang (2000) develops a new method for the estimation of radial distortion co-efficients based on radial distortion snakes. These snakes perform like normal deformable contours. The only difference is that their attitude is globally connected through a constant model of image radial distortion. Figure 2.19 shows different snake configurations. Kang (2000) claims his method could be extended to address the problems of tangential distortion and a non-central principle point. Also the whole process of removing radial distortion using edge detection and linking proceeded by hypothesis testing could be automatic. This step is followed by a robust estimator to reject outliers.

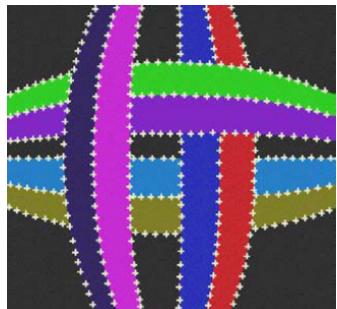
Devernay and Faugeras (2001) introduces an algorithm that is plumb-lined based. Figure 2.20 shows sample results of their undistortion procedure. Following are the steps of their algorithm:

- Extract edges from an image.
- Approximate polygonal image with large tolerance.
- Transform these edges into segments to find possible lines and the parameters of distortion model.

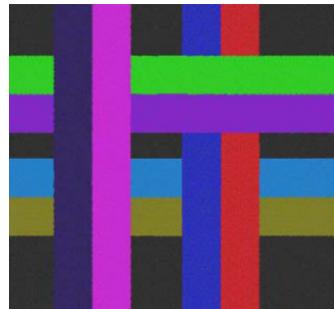
Thormählen et al. (2003) introduces a new method that does not require any calibration pattern or any information about camera parameters. The authors claim that their method



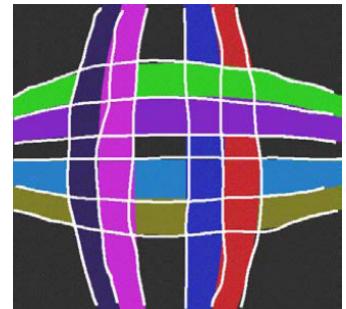
(a)



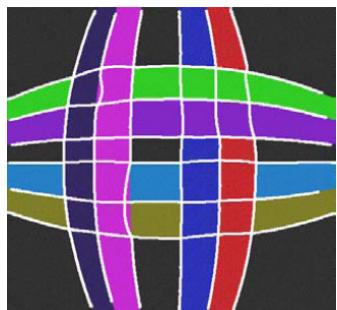
(b)



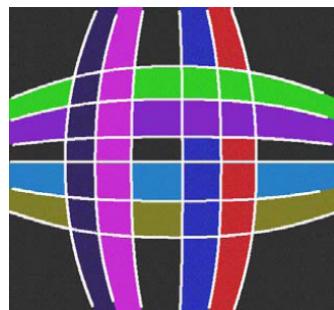
(c)



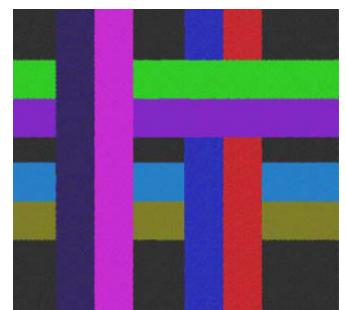
(d)



(e)

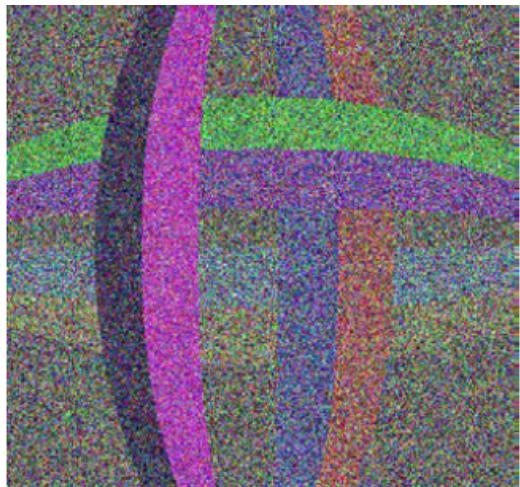


(f)

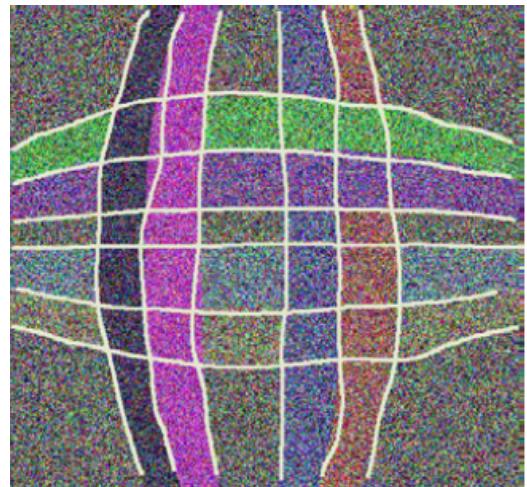


(g)

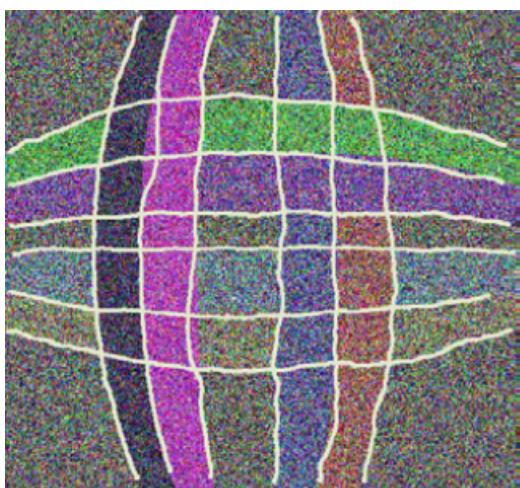
Figure 2.13: Synthetic images with no noise. (a) Original image. (b) Manually selected points. (c) Undistorted image. (d) Manually drawn approximate lines. (e) Results from conventional snake algorithm. (f) Results from radial distortion snake algorithm. (g) Undistorted images (e) and (f). Reprinted from Kang (1997).



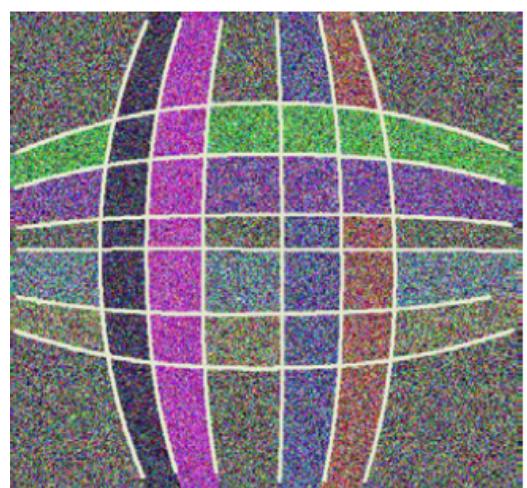
(a)



(b)



(c)



(d)

Figure 2.14: Synthetic images with Gaussian noise with standard deviation of 100.
(a) Original image. (b) Manually drawn approximate lines. (c) Results of conventional snake algorithm. (d) Results of radial distortion snake algorithm. Reprinted from Kang (1997).

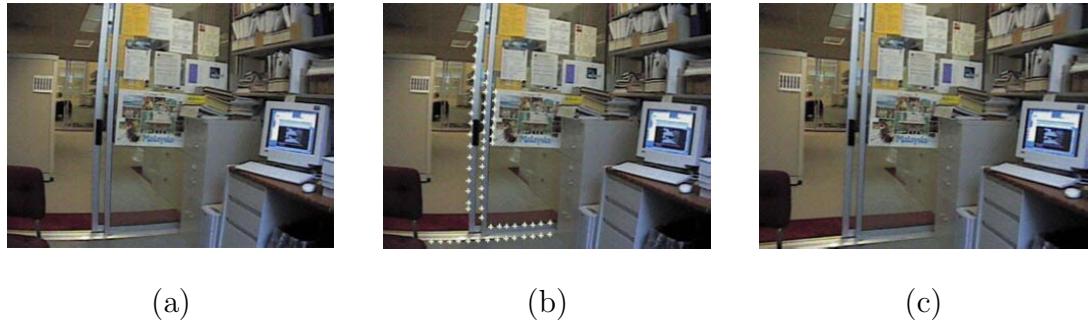


Figure 2.15: Manual method is based on manual selection of points that are used to compute the radial distortion parameters. (a) Original image. (b) Manually drawn lines. (c) Undistorted image. Reprinted from Kang (1997).

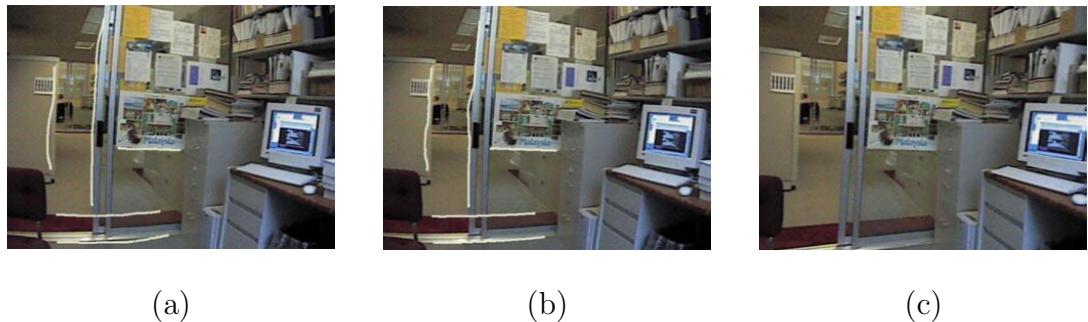


Figure 2.16: Conventional snake method allows user to draw snake contour on the image that is equivalent to a line in the world. (a) Chosen snakes using conventional snake method. (b) Optimized snake contour. (c) Undistorted image. Reprinted from Kang (1997).

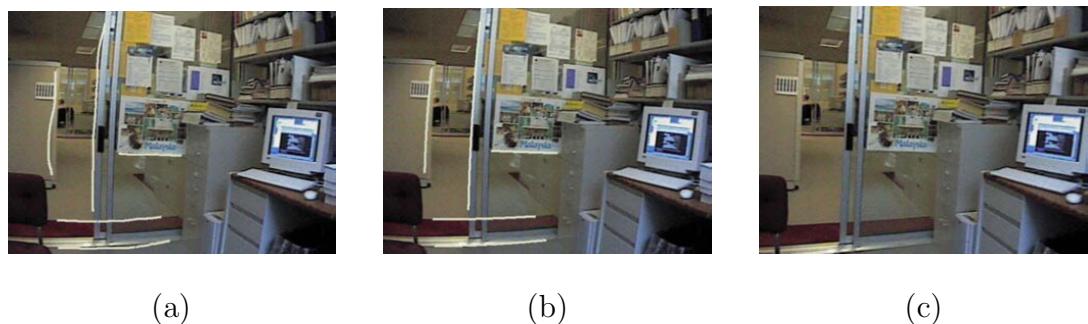


Figure 2.17: Radial distortion snake method is based on independent and conventional snake contour. (a) Chosen snakes using radial distortion snake method. (b) Optimized snake contour. (c) Undistorted image. Reprinted from Kang (1997).

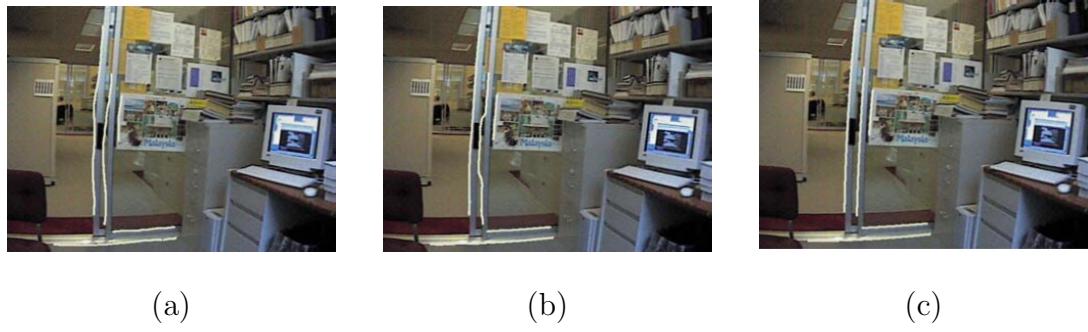


Figure 2.18: Comparing the two snake implementation. (a) Initial and optimized snake contour for comparing the two snake algorithms. (b) Conventional snake. (c) Radial distortion snake. Reprinted from Kang (1997).

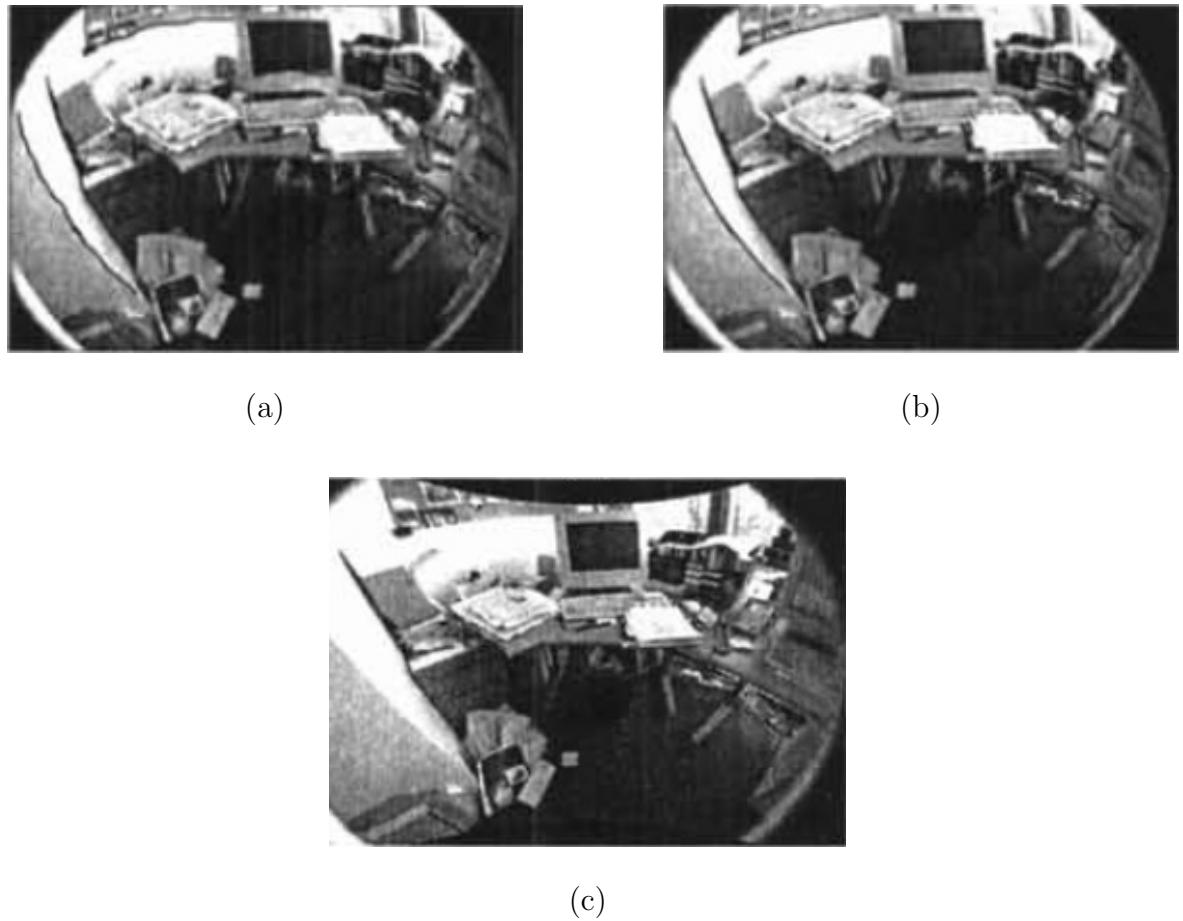


Figure 2.19: Radial distortion removal example. (a) Initial hand-drawn snake on black line super imposed on column corner at left side of image. (b) Optimized snake contour. (c) Undistorted image. Reprinted from Kang (2000).

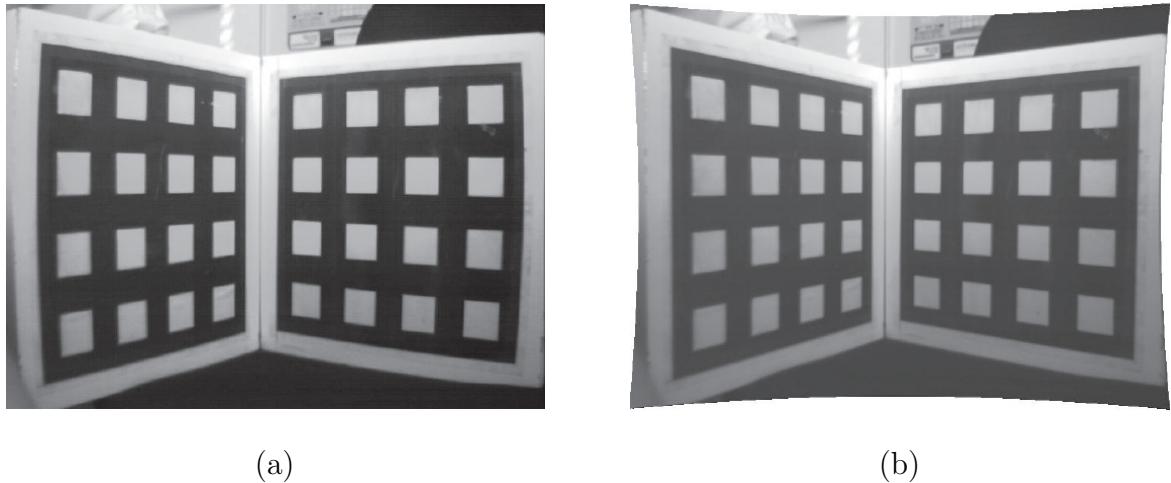


Figure 2.20: Radial distortion removal example based on Tsai (1987)'s image. (a) Distorted image. (b) Undistorted image. Reprinted from Devernay and Faugeras (2001).

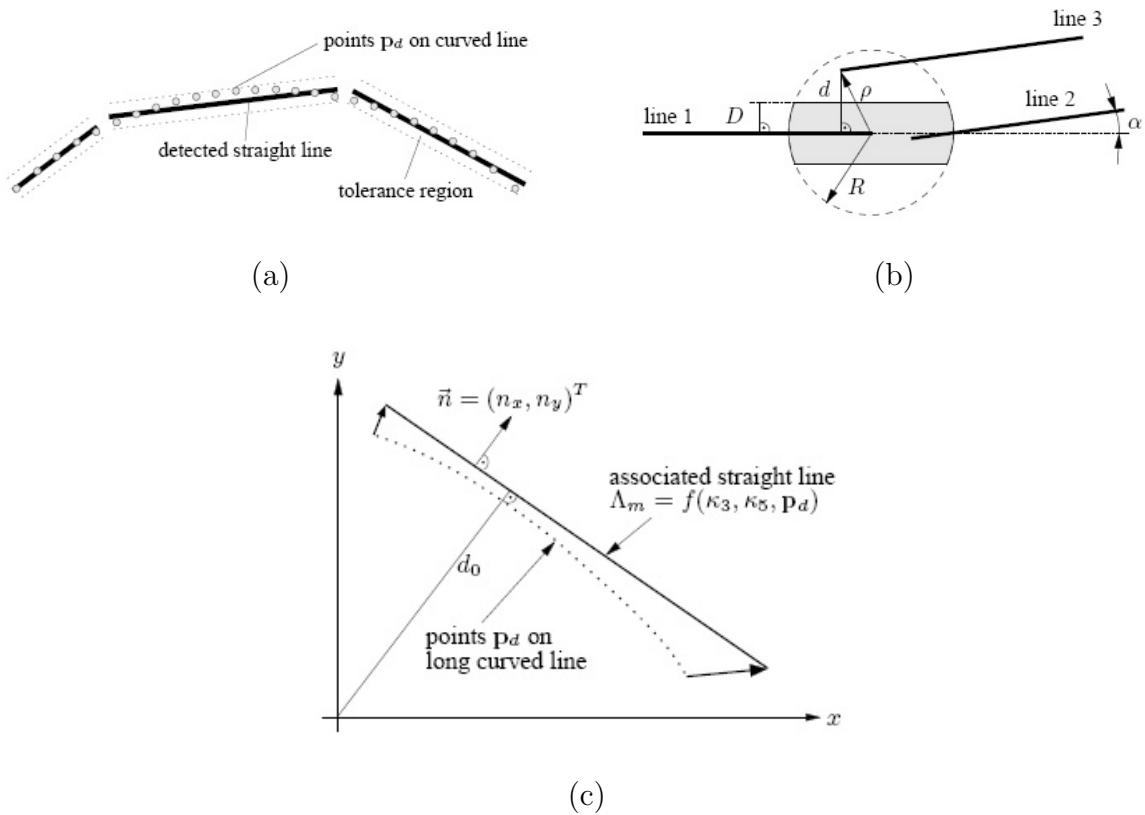
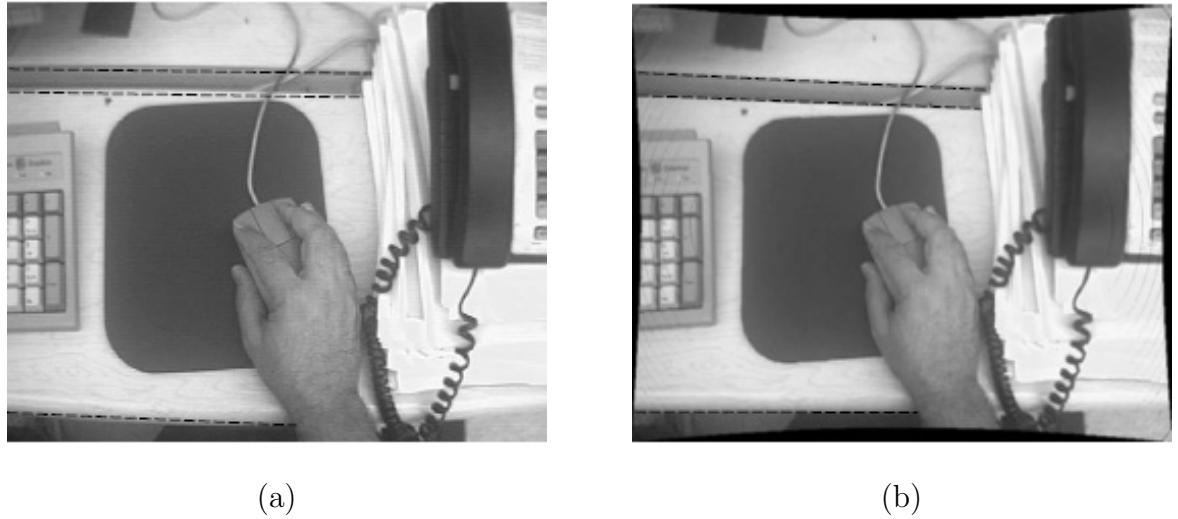


Figure 2.21: Thormählen et al. (2003)'s procedure for the undistorted lines. (a) Detection of points p_d on curved lines. (b) From segments of curved lines, linkage of straight lines. (c) Points p_d on the curved line are mapped to points p_u on the associated straight line Λ_m . It is a function of k_3 , k_5 and the points p_d . Reprinted from Thormählen et al. (2003).



(a)

(b)

Figure 2.22: Undistorted image using El-Melegy and Farag (2003)'s method. (a) Distorted image. (b) Undistorted image. Reprinted from El-Melegy and Farag (2003).

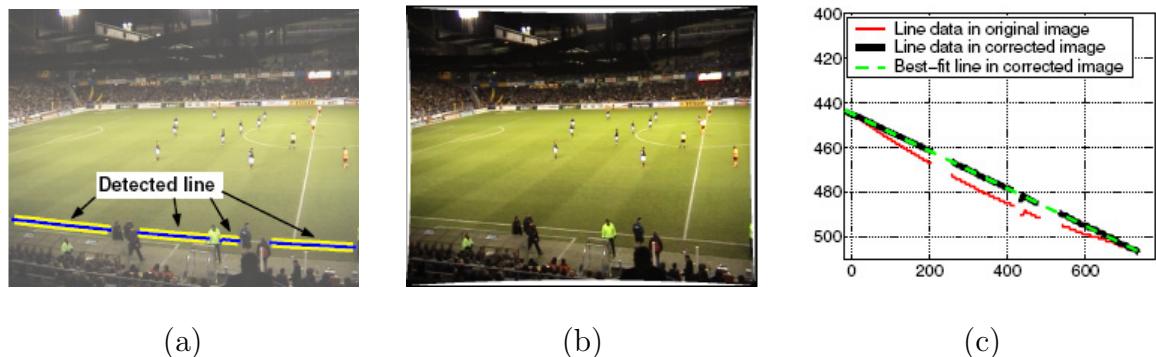


Figure 2.23: Sample results of Strand and Hayman (2005)'s method. (a) Original image. (b) Undistorted image. (c) Figures of line before and after undistortion. Reprinted from Strand and Hayman (2005).

tackle the problem of real curves remain curves after undistortion. Figure 2.21 shows the main steps of their algorithm. Following are the main steps of their algorithm:

- Detect points on curved lines.
- Link the curved lines.
- Perform outlier elimination.
- Do parameter estimation of the inverse radial distortion model.

El-Melegy and Farag (2003) present an automatic approach to non-metric calibration method based on the robust the-least-median-of-squares (LMedS) estimator. Figure 2.22 shows the performance of his method.

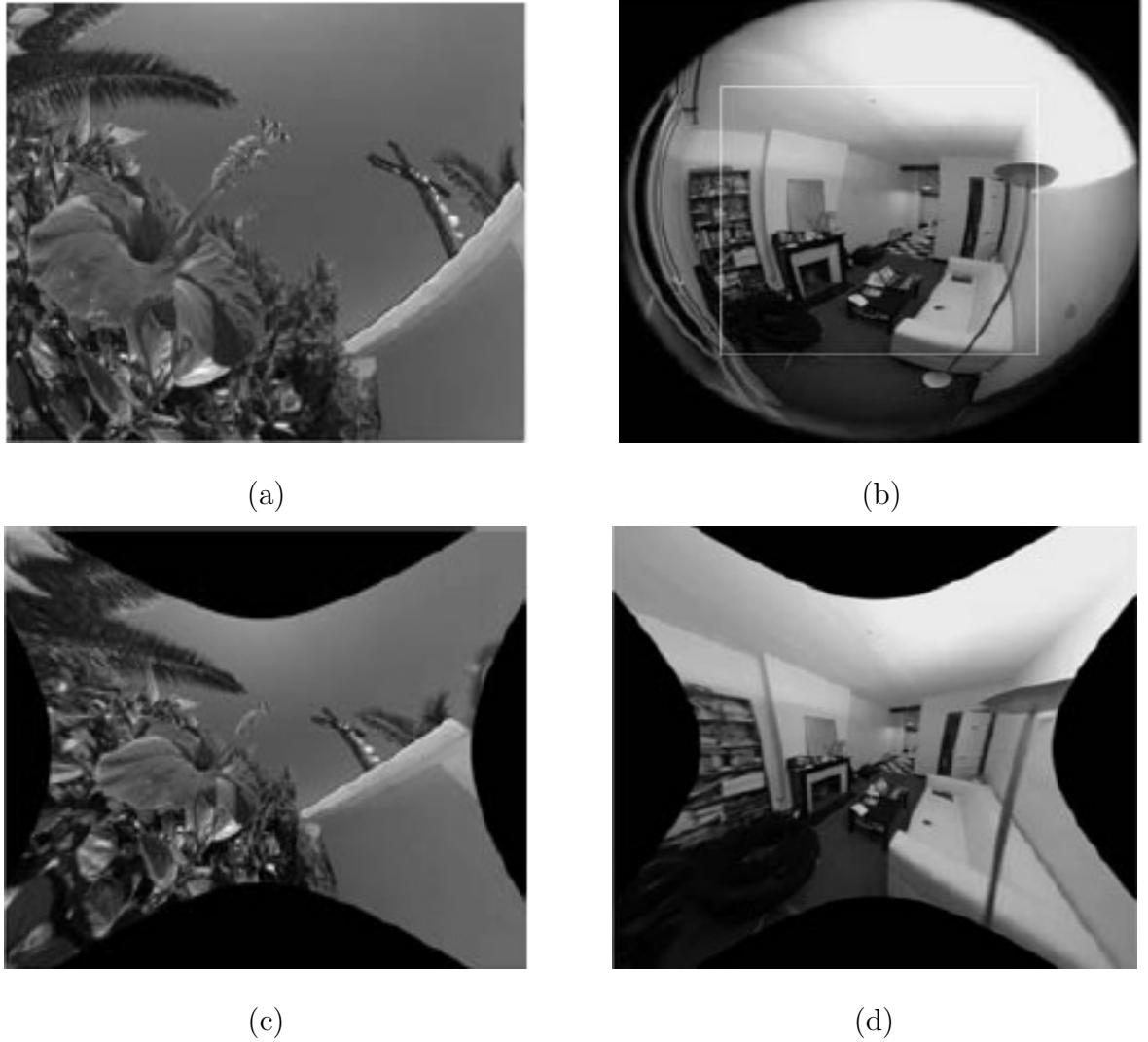


Figure 2.24: Undistorted images using Wang et al. (2009)'s method. (a,b) Distorted images. (c,d) Undistorted images using the single parameter division model based that is based on circle fitting. Reprinted from Wang et al. (2009).

Strand and Hayman (2005) introduce a method which is based on the single parameter division model. They say that distorted straight lines form a circle. This reduces the problem of estimation of distortion parameters to circle fitting. The authors introduce a number of techniques including closed-form solutions and iterative method. Some of the closed-form solutions give excellent results. The iterative method gives optimal results in case of Gaussian noise in the actual image plane. Like most method, this one assumes known aspect ratio and zero skew. If the aspect ratio is non-unity, then the circle becomes an ellipse with principle axes in the x and y directions. It should not be cumbersome to derive linear or iterative algorithms after relaxing the constraints. Another improvement could extend the method based on the single parameter division model by including two or more parameters. Figure 2.23 (a), Figure 2.23 (b), and Figure 2.23 (c) show an original image, corresponding undistorted image, and a line before and after undistortion respectively.

Wang et al. (2009) prove mathematically that under the single parameter division model a

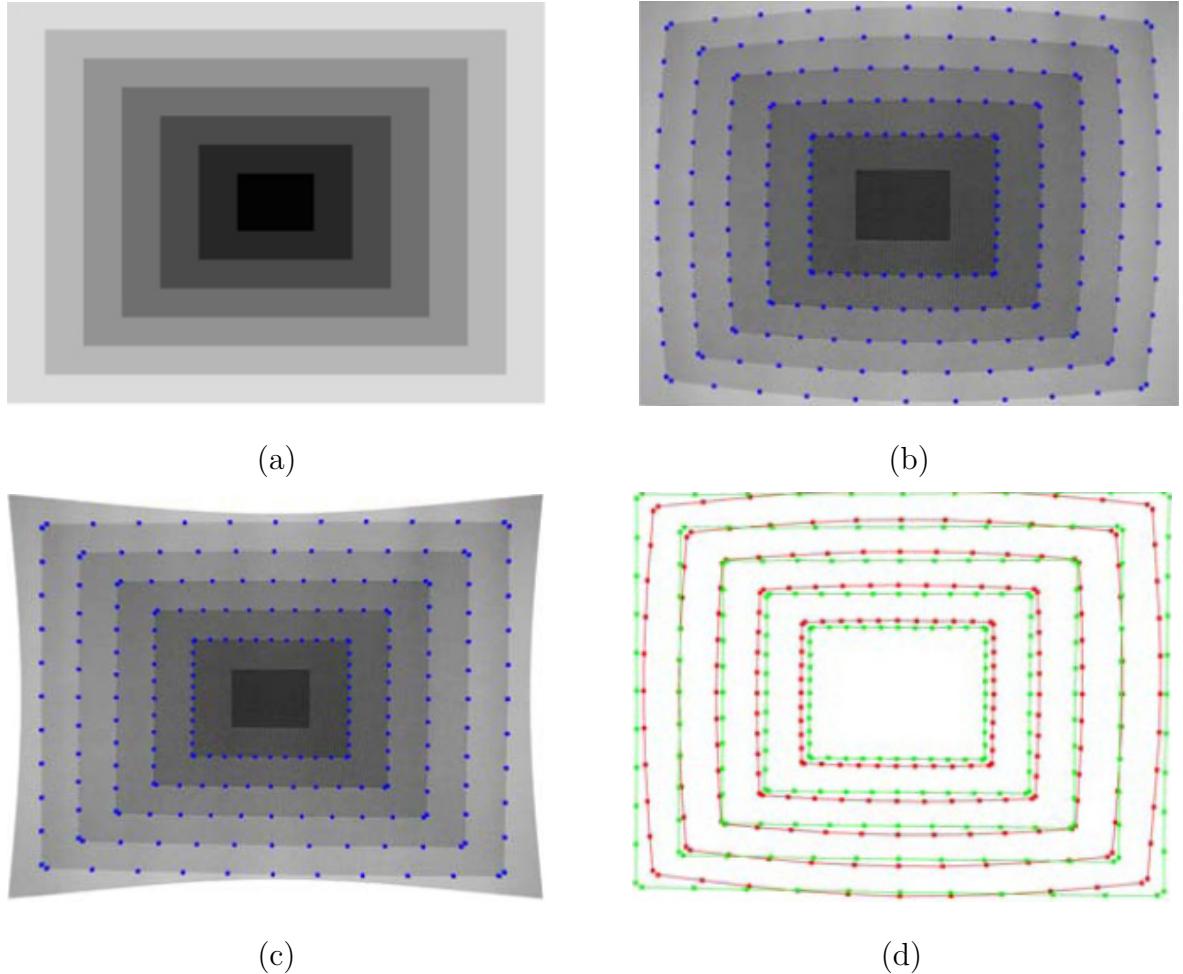


Figure 2.25: Comparison of distorted and undistorted points. (a) Synthetic image. (b) Distorted synthetic image. Blue dots represent distorted points. (c) Undistorted image. Blue dots represent undistorted points. (d) Comparison of distorted and undistorted lines. Red dots represent distorted points and green dots represent corresponding undistorted points. Reprinted from Alvarez et al. (2009).

distorted line becomes a circular arc. Also, they devise an algorithm that simultaneously estimates the radial distortion parameter and the distortion center. They propose different solutions when there is only one line available in the image, two lines available in the image, and if the pixels are non-square. Their algorithm is non-iterative, but their method seems to be manual. Figure 2.24 (a,b) show original images and Figure 2.24 (a) and (b) show corresponding undistorted images, respectively.

Alvarez et al. (2009) introduce an algebraic approach to eliminate radial distortion based on rectification of lines in an image. Their cost function is based on a 4 degree polynomial in several variables. The distortion parameters are obtained by minimizing the cost function. They assume that the distortion center is the image center. They have developed a Web demo, in which a user can select points on one or more lines and undistort an image. Figure 2.25 shows the sample results of their method.



(a)



(b)



(c)



(d)

Figure 2.26: Undistorted images at different focal lengths. (a) Distorted image at focal length 45.16 mm. (b) Distorted image at focal length 156.55 mm. (c) Undistorted image at focal length 45.16 mm. (d) Undistorted image at focal length 156.55 mm. The zoom focus is unknown and is estimated. Reprinted from León et al. (2011).

León et al. (2011) develop a new method mathematical models for different variations of lens distortion models based on a zoom camera. The models depend on polynomial approximation to compensate for the variation of the radial distortion parameters using the range of zoom. Also their method depends on minimization of a global error energy, which measures the distance between distorted points and undistorted points. Figure 2.26 (a) and (c) show the distorted images at focal length of $f = 45.16$ mm and $f = 156.55$ mm. Figure 2.26 (b) and (d) show corresponding undistorted images at different focal length respectively.

2.4 Analysis of state of the art of plumb-line based methods

The main limitations of plumb-line based methods are that straight lines must be visible in the image and that images of actual curved lines may disrupt estimation. Some methods address these issues simply by utilizing human supervision to select the lines (see, e.g., (Brown, 1971; Swaminathan & Nayar, 2000; Alvarez et al., 2009)). But when human supervision is not used, plumb-line methods depend critically on the robustness and accuracy of the line detection algorithms. Some plumb-line approaches do not use all available lines for distortion estimation despite the fact that additional lines could minimize estimation error (Thormählen et al., 2003; Strand & Hayman, 2005; Wang et al., 2009), or assume the distortion center as the center of the image (Kukelova & Pajdla, 2011; Alvarez et al., 2009; Brauer-Burchardt & Voss, 2001; Strand & Hayman, 2005; Fitzgibbon, 2001), which is in contrast to some researchers' recommendations (Hartley & Kang, 2007; Tavakoli & Pourreza, 2010).

The Devernay and Faugeras (Devernay & Faugeras, 2001) method is the only existing method that overcomes all of these limitations. However, it requires a complex process of polygonal approximation of the distorted lines. As we shall see, the distorted line detection process can be dramatically simplified by using an alternative distortion model.

2.5 Conclusion

We propose a new method based on the plumb-line approach that addresses the aforementioned limitations. The method works from a single image if the image contains a sufficient number of distorted straight lines. It does not require a calibration pattern or human intervention. We use Fitzgibbon's division model of radial distortion (Fitzgibbon, 2001) with a single parameter. Our estimator is similar to that of Strand and Hayman (Strand & Hayman, 2005) and Wang et al. (Wang et al., 2009) in that we estimate the parameters of the distortion model from the parameters of circular arcs identified in the distorted image, based on the fact that distorted straight lines can be modeled as circular under the division model (Wang et al., 2009; Strand & Hayman, 2005; Barreto & Daniilidis, 2005). Our contribution is to make the process fully automatic and robust to outliers using a two-step random sampling process. For the first step, we introduce a sampling algorithm to search the input image for subsequences of contour pixels that can be modeled as circular arcs. For the second step, we introduce a sampling algorithm that finds the distortion parameters consistent with the largest number of arcs. Based on these parameters, we undistort the input image. Some preliminary results from our method have previously appeared in a conference paper (Bukhari & Dailey, 2010).

In this dissertation, to evaluate the new algorithm, we perform a comprehensive quantitative study of its performance on distorted synthetic images and provide extensive examples of its ability to remove distortion from a large, challenging set of real images taken from Web sites and previous papers on distortion estimation. We find that the algorithm performs very well, with excellent reconstruction of the original synthetic images even under severe barrel distortion and pincushion distortion. We also find that the method is able to eliminate nearly all of the visible distortion in the real images, including those acquired with wide angle and fish-eye lenses. Finally, we perform a direct comparison of our method with that of Alvarez et al. (Alvarez et al., 2009), the only researchers who have provided a publicly accessible implementation of their method, on synthetic images. The Alvarez et al. method exploits user assistance in identifying points on straight lines, but we nevertheless find that our fully automatic method provides superior reconstruction of the original undistorted image. Our method is thus a practical solution to the important problem of radial distortion estimation and removal.

2.6 The Canny edge detector

One problem with OpenCV’s Canny edge detector is that we have to manually specify a low gradient threshold and a high gradient threshold. We observed that sometimes the default gradient thresholds provided by OpenCV would not give us optimal edges. The quality of detected edges largely depend on how we select these thresholds. Matlab’s Canny edge detector automatically selects a low gradient threshold and a high gradient threshold and we find these thresholds work reasonably well across different images. For edge extraction, we modified OpenCV’s Canny edge detector to automatically select a low gradient threshold and a high gradient threshold based on a cumulative histogram of magnitudes, similar to the Matlab implementation. For edge extraction, we modified OpenCV’s Canny edge detector to automatically select a low gradient threshold and a high gradient threshold based on a cumulative histogram of magnitudes, similar to the Matlab implementation of Canny’s method (MATLAB, 2009). See Canny’s method (MATLAB, 2009) for details.

2.7 Random Sample and Consensus (RANSAC)

Fischler and Bolles (1981) develop an algorithm that extracts inliers and removes outliers. Outliers have also been the primary weakness of least squares methods. Least squares methods fit a model using all points. They are severely affected by outliers. Figure 2.27 distinguishes the difference between least squares and RANSAC method. See Figure 2.28 for a flow chart. Following are the main steps of RANSAC algorithm:

- Randomly select minimal set of points to fit a specific model.
- Find all points within some distance threshold of the model. Name this set of points the consensus set. The size of this set is the model’s support.
- Repeat for N samples. Return the model with the largest consensus set.

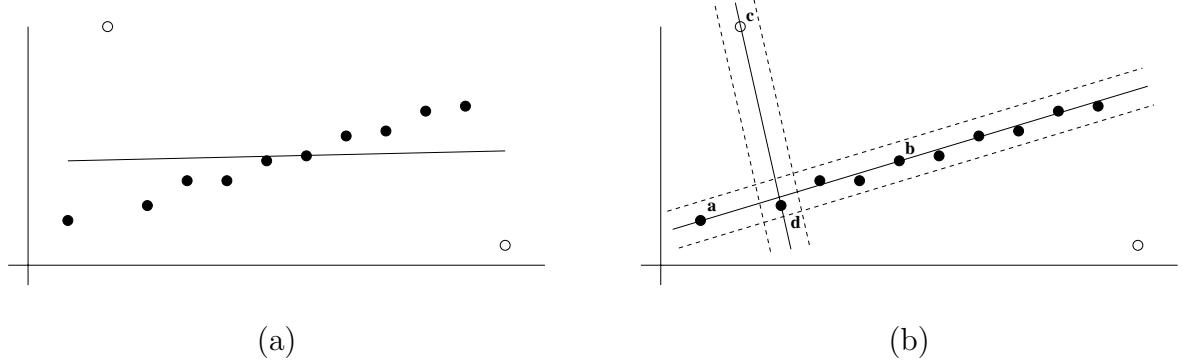


Figure 2.27: Least squares vs. RANSAC model fitting including two outliers. (a) Result of least squares based line fitting. (b) Result of RANSAC based line fitting. Reprinted from Hartley and Zisserman (2004).

2.8 Circle fitting methods

Radial distortion bends straight lines into circular arcs, violating the main invariance preserved in the pinhole camera model, that straight lines in the world map to straight lines in the image plane. When lines are distorted in an image then they become circles under the single parameter division model. Also, see section 3.1.2 for a mathematical prove. We discuss both algebraic and geometric circle fitting method in coming sections. For a comparative summary of the different circle fitting algorithms, refer to Table 2.1, and for more detailed discussion of the methods, refer to Chernov (Chernov, 2010).

2.9 Algebraic circle fitting methods

In this section we discuss different circle algebraic circle fitting methods, specifically:

- Pratt (Pratt, 1987).
- Taubin (Taubin, 1991).
- Kukush-Markovsky-van-Huffel (KMvH) (Kukush, Markovsky, & Van Huffel, 2004).

2.9.1 Pratt and Taubin circle fitting methods

Both Pratt and Taubin use four parameters to specify a circle:

$$a(x^2 + y^2) + bx + cy + d = 0,$$

with $a \neq 0$.

The center of the circle is $(-\frac{b}{2a}, -\frac{c}{2a})$ and the radius is given by $r = \sqrt{(-\frac{b}{2a})^2 + (-\frac{c}{2a})^2 - \frac{d}{a}}$.

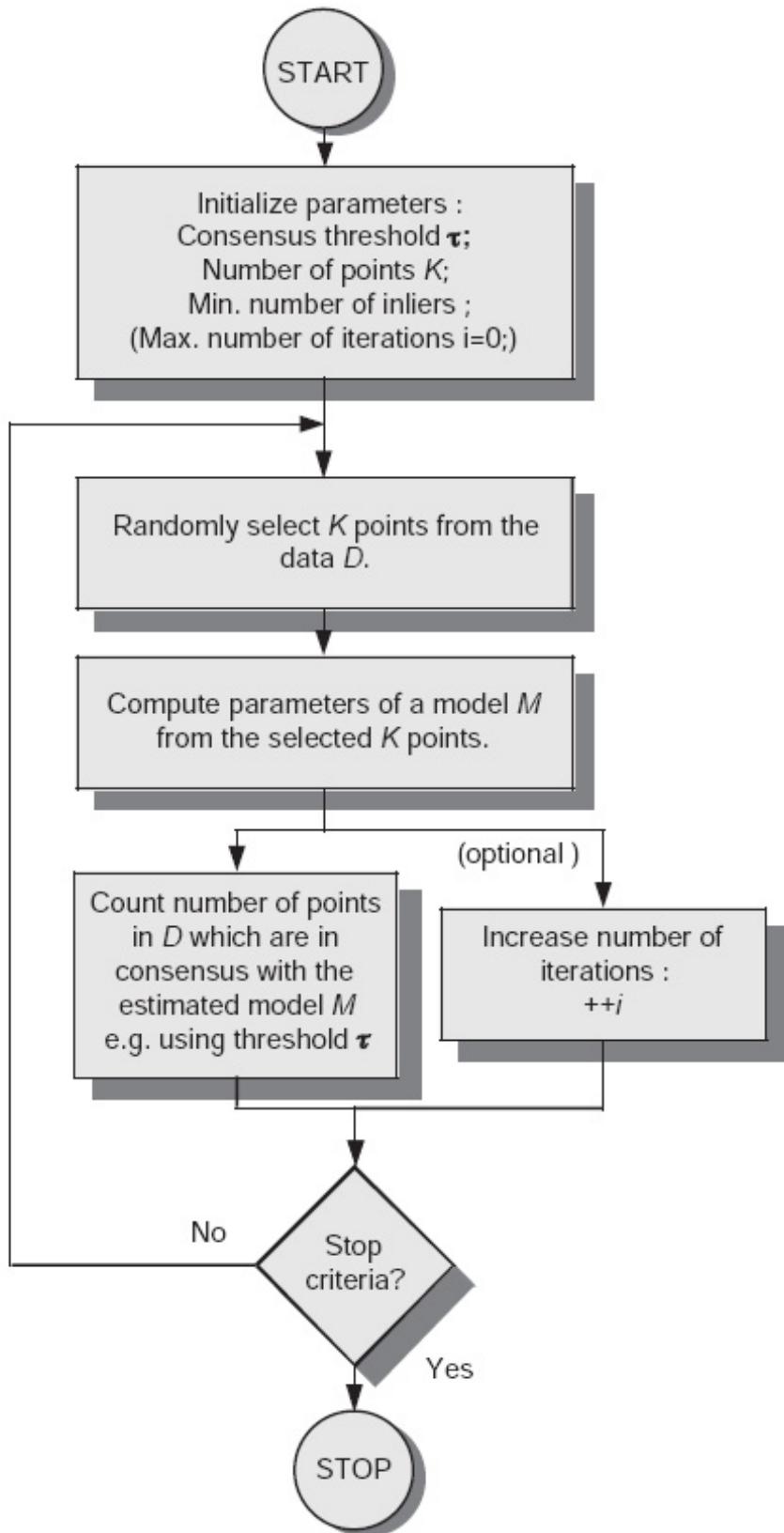


Figure 2.28: Flow chart of the RANSAC algorithm. Reprinted from (Cyganek & Siebert, 2009)

The Pratt method minimizes the objective function

$$\Omega(a, b, c, d) = \sum_{i=1}^N a(x_i^2 + y_i^2) + bx_i + cy_i + d^2,$$

subject to the constraint that $b^2 + c^2 - 4ad = 1$, to ensure that the parameterized equation represents an actual circle. The Taubin method minimizes the same objective function as the Pratt method, but imposes the constraint $4a^2\bar{z} + 4ab\bar{x} + 4ac\bar{y} + b^2 + c^2 = 1$, where \bar{x} is the mean of the sample points' x coordinates, \bar{y} is the mean of the sample points' y coordinates, and $\bar{z} = \frac{1}{N} \sum_{i=1}^N (x_i^2 + y_i^2)$. The additional constraint improves the convergence of the optimization (Chernov, 2010).

2.9.2 Kukush-Markovsky-van-Huffel (KMvH)

We use one other algebraic method, the Kukush-Markovsky-van-Huffel (KMvH) consistent conic fitting method (Kukush et al., 2004), which minimizes the objective function $\Omega(a, b, c, d) = \mathbf{A}^T \mathbf{M} \mathbf{A}$, where $\mathbf{A} = [a \ b \ c \ d]^T$ and \mathbf{M} is an unbiased estimate of the data covariance matrix. The method guarantees convergence to the true parameters as the number of sampled points approaches infinity.

2.10 Geometric circle fitting methods

In this section we discuss different circle geometric circle fitting methods, specifically:

- Generic LM.
- Reduced-LM.
- Chernov and Lesort LM (Chernov-LM).
- Trust-Region-LM.

2.10.1 Generic LM

The gold standard objective function for circle fitting is

$$\Omega(x_c, y_c, r) = \sum_{i=1}^N d(x_i, y_i, x_c, y_c, r)^2, \quad (\text{Equation 2.1})$$

where (x_c, y_c) is the center of the circle, r is its radius, and $d(x, y, x_c, y_c, r)$ is the orthogonal distance of the measured point (x, y) to the hypothetical circle. N is the number of pixels in a inlier contour. Since there is no closed-form solution for minimizing this objective function (Al-Sharadqah & Chernov., 2009), we use an initial guess and the Levenberg-Marquardt

(LM) least squares method to find a local minimum. To obtain the initial guess, we use a variety of methods as detailed in the next section.

2.10.2 Reduced-LM

In the Reduced-LM method (Chernov, 2010), only two parameters are adjusted; i.e., x_c and y_c (the coordinates of the center of the circle). The method minimizes the objective function

$$\Omega(x_c, y_c) = \sum_{i=1}^N \left[\bar{r} - \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \right]^2,$$

where $\bar{r} = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$.

2.10.3 Chernov and Lesort LM

The Chernov and Lesort LM method (Chernov & Lesort, 2005) guarantees convergence to a minimum of the objective function from any initial guess. They redefine the circle parameters with respect to Equation 2.1 as $A = -\frac{b}{2a}$, $B = -\frac{c}{2a}$, and $R^2 = \frac{b^2+c^2-4ad}{4a^2}$. Their objective function is

$$\Omega(a, b, c, d) = 2 \frac{P_i}{1 + \sqrt{1 + 4aP_i}},$$

where $P_i = a(x_i^2 + y_i^2) + bx_i + ci + d$.

2.10.4 Trust-Region-LM

The Trust-Region-LM method (Moré, 1978) guarantees proper initialization of LM's learning rate or control parameter λ and also provides an efficient rule for updating λ 's value.

We coded all the geometric and algebraic circle fitting methods in C++ using OpenCV (Bradski, 2000), with reference to Chernov's (Chernov, 1997) MATLAB implementations.

Table 2.1: Comparison of different circle fittings methods. LM = Levenberg Marquardt, KMvH = Kukush Markovsky van Huffel; IURAT = Invariant Under Rotations And Translations. $r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$, $\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i$; $A = \pm \frac{1}{2R}$, $B = -2Aa$, $C = -2Ab$, $D = \frac{B^2+C^2-1}{4A}$; $\bar{z} = \frac{1}{N} \sum_{i=1}^N (x_i^2 + y_i^2)$. (x_c, y_c) is the center and R is the radius of the circle. \mathbf{A} is a vector representing the parameters of a circle. Initial guess = IG, Outliers resistant = OR, Geometric = GC, Analytic = AC, IURAT = IT.

Method	IG	OR	GC	AC	IT	Objective function	Comments
Ransac	Y	Y	N	N	Y	None	Simple calculation of circle parameters from three points.

Continued ...

Table 2.1: Comparison of different circle fittings methods. LM = Levenberg Marquardt, KMvH = Kukush Markovsky van Huffel; IURAT = Invariant Under Rotations And Translations. $r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$, $\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i$; $A = \pm \frac{1}{2R}$, $B = -2Aa$, $C = -2Ab$, $D = \frac{B^2 + C^2 - 1}{4A}$; $\bar{z} = \frac{1}{N} \sum_{i=1}^N (x_i^2 + y_i^2)$. (x_c, y_c) is the center and R is the radius of the circle. \mathbf{A} is a vector representing the parameters of a circle. Initial guess = IG, Outliers resistant = OR, Geometric = GC, Analytic = AC, IURAT = IT (continued).

Method	IG	OR	GC	AC	IT	Objective function	Comments
Generic-LM	Y	N	Y	N	Y	$\Omega(x_c, y_c, R) = \sum_{i=1}^N (r_i - R)^2$	Properly initializing the learning rate is cumbersome (Chernov, 2010).
Chernov-LM	Y	N	Y	N	N	$\Omega(a, b, c, d) = 2 \frac{P_i}{1 + \sqrt{1 + 4aP_i}}$, where $P_i = a(x_i^2 + y_i^2) + bx_i + cy_i + d$	Guaranteed convergence to a minimum of the objective function but cost per iteration is 4 times higher than any other geometric circle fitting method. Sensitive to the choice of coordinate system (Chernov & Lesort, 2005).
Reduced-LM	Y	N	Y	N	Y	$\Omega(x_c, y_c) = \sum_{i=1}^N (\bar{r} - \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2})^2$, where $\bar{r} = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$	Objective function based only on the circle center. Faster than generic LM but more sensitive to bad initial guesses (Chernov, 2010).
Trust-Region-LM	Y	N	Y	N	Y	$\Omega(x_c, y_c, R) = \sum_{i=1}^N (r_i - R)^2$	Good rule for proper initialization and updating of the learning rate (Moré, 1978).
Pratt	N	N	N	Y	Y	$\Omega(a, b, c, d) = \sum_{i=1}^n a(x_i^2 + y_i^2) + bx_i + cy_i + d^2$, subject to the constraint $b^2 + c^2 - 4ad = 1$	Estimate converges to true parameters in case of small noise (Pratt, 1987).
Taubin	N	N	N	Y	Y	$\Omega(a, b, c, d) = \sum_{i=1}^n a(x_i^2 + y_i^2) + bx_i + cy_i + d^2$, subject to the constraint $4a^2\bar{z} + 4ab\bar{x} + 4ac\bar{y} + b^2 + c^2 = 1$	Can be generalized to ellipses and other geometric curves (Taubin, 1991). Less biased than Pratt's method. Considered statistically more accurate than Pratt (Al-Sharadqah & Chernov., 2009).

Continued ...

Table 2.1: Comparison of different circle fittings methods. LM = Levenberg Marquardt, KMvH = Kukush Markovsky van Huffel; IURAT = Invariant Under Rotations And Translations. $r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$, $\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i$; $A = \pm \frac{1}{2R}$, $B = -2Aa$, $C = -2Ab$, $D = \frac{B^2 + C^2 - 1}{4A}$; $\bar{z} = \frac{1}{N} \sum_{i=1}^N (x_i^2 + y_i^2)$. (x_c, y_c) is the center and R is the radius of the circle. \mathbf{A} is a vector representing the parameters of a circle. Initial guess = IG, Outliers resistant = OR, Geometric = GC, Analytic = AC, IURAT = IT (continued).

Method	IG	OR	GC	AC	IT	Objective function	Comments
KMvH	N	N	N	Y	Y	$\Omega(a, b, c, d) = \mathbf{A}^T \mathbf{M} \mathbf{A}$, where $\mathbf{A} = [a \ b \ c \ d]^T$ and \mathbf{M} is an unbiased estimate of the data covariance matrix.	Guaranteed convergence to the true parameters as the number of sampled points approaches infinity (Kukush et al., 2004; Chernov, 2010).

Chapter 3

Methodology

This chapter explains my research methodology, based on my objectives. The primary goal of this research is to design and implement an automatic radial distortion estimation method based on the plumb-line approach. This chapter is composed of the following sections: Mathematical Model and Robust Radial Distortion Estimation.

3.1 Mathematical Model

In this section, we outline the mathematical model of radial distortion assumed in the rest of the dissertation and show how to estimate the parameters of this model.

3.1.1 Distortion model

The most commonly used radial distortion model is the *even-order polynomial model*

$$\begin{aligned}x_u &= x_d(1 + \lambda_1 r_d^2 + \lambda_2 r_d^4 + \lambda_3 r_d^6 + \dots) \\y_u &= y_d(1 + \lambda_1 r_d^2 + \lambda_2 r_d^4 + \lambda_3 r_d^6 + \dots),\end{aligned}\quad (\text{Equation 3.1})$$

where (x_u, y_u) and (x_d, y_d) are the corresponding coordinates of an undistorted point and a distorted point, respectively. r_d is the Euclidean distance of the distorted point to the distortion center. If the distortion center is the origin of the distorted image, we can simply write

$$r_d^2 = x_d^2 + y_d^2. \quad (\text{Equation 3.2})$$

However, if (x_0, y_0) is the center of distortion (in the distorted image), we write

$$r_d^2 = (x_d - x_0)^2 + (y_d - y_0)^2 \quad (\text{Equation 3.3})$$

and replace x_d and y_d in Equation 3.1 with $(x_d - x_0)$ and $(y_d - y_0)$, respectively. In the model, $x_0, y_0, \lambda_1, \lambda_2, \lambda_3, \dots$ are the *distortion parameters*, which must be estimated from image measurements.

There have been objections to the even-order polynomial model. According to Wang et al. (2009), the model performs well for small distortion, but for severe distortion, a prohibitively large number of non-zero distortion parameters are required.

Fitzgibbon (Fitzgibbon, 2001) proposes an alternative model, the *division model*, as a more accurate approximation to the typical camera's true undistortion function:

$$\begin{aligned} x_u &= \frac{x_d}{1 + \lambda_1 r_d^2 + \lambda_2 r_d^4 + \dots} \\ y_u &= \frac{y_d}{1 + \lambda_1 r_d^2 + \lambda_2 r_d^4 + \dots}. \end{aligned}$$

The division model is preferred over the polynomial model because it requires fewer terms than the polynomial model in case of severe distortion (Wang et al., 2009). It is also slightly easier to work with; inverting the single-parameter division model, for example, requires solution of a polynomial of degree two, whereas inverting the single-parameter polynomial model leads to a polynomial of degree three. In our work, we use the single-parameter division model (fixing $\lambda_2 = \dots = 0$), because for most cameras, a single term is sufficient (Devernay & Faugeras, 2001; Fitzgibbon, 2001; Wang, Shi, Zhang, & Liu, 2008; Wang et al., 2009). When the center of distortion is the origin, we can write the single-parameter division model in the form

$$\begin{aligned} x_u &= \frac{x_d}{1 + \lambda r_d^2} \\ y_u &= \frac{y_d}{1 + \lambda r_d^2} \end{aligned} \tag{Equation 3.4}$$

with r_d^2 defined according to Equation 3.2. When the center of distortion is not the origin, we can write the single-parameter division model in the form

$$\begin{aligned} x_u &= x_0 + \frac{x_d - x_0}{1 + \lambda r_d^2} \\ y_u &= y_0 + \frac{y_d - y_0}{1 + \lambda r_d^2} \end{aligned} \tag{Equation 3.5}$$

with r_d^2 defined according to Equation 3.3. Strand and Hayman (Strand & Hayman, 2005) find that for the typical case of relatively small barrel distortion (small negative values for λ), the single-parameter division model is highly correlated with the single-parameter polynomial model.

3.1.2 Distortion of a line under the single-parameter division model

Wang et al. (2009) show that under the single-parameter division model, the distorted image of a straight line is a circular arc. However, they use the slope- y -intercept form of the equation of a line, which we avoid due to its inability to model vertical lines and its undesirable numerical properties. However, it is also easy to show that the general line

$$ax_u + by_u + c = 0 \tag{Equation 3.6}$$

is also imaged as a circular arc under the single parameter division model. To avoid the degenerate case $a = b = 0$, we impose the constraint that $a^2 + b^2 > 0$. (When convenient we will further assume the line parameters are normalized so that $a^2 + b^2 = 1$.) First assume that the distortion is the image center. By substituting the image coordinates from Equation 3.4 into Equation 3.6

$$a\left(\frac{x_d}{1+\lambda r_d^2}\right) + b\left(\frac{y_d}{1+\lambda r_d^2}\right) + c = 0$$

$$\implies ax_d + by_d + c(1 + \lambda r_d^2) = 0$$

replacing r_d^2 by its definition from Equation 3.2

$$\implies ax_d + by_d + c(1 + \lambda x_d^2 + \lambda y_d^2) = 0$$

$$\implies c\lambda x_d^2 + c\lambda y_d^2 + ax_d + by_d + c = 0$$

$$\implies x_d^2 + y_d^2 + \frac{a}{c\lambda}x_d + \frac{b}{c\lambda}y_d + \frac{1}{\lambda} = 0 \quad (\text{Equation 3.7})$$

Equation 3.7 is the equation of a circle, when image center is the distortion center. Now assume that the distortion is not the image center. Replace x_d and y_d in Equation 3.7 with $(x_d - x_0)$ and $(y_d - y_0)$, respectively

$$\implies (x_d - x_0)^2 + (y_d - y_0)^2 + \frac{a}{c\lambda}(x_d - x_0) + \frac{b}{c\lambda}(y_d - y_0) + \frac{1}{\lambda} = 0$$

$$\implies x_d^2 + x_0^2 - 2x_d x_0 + y_d^2 + y_0^2 - 2y_d y_0 + \frac{a}{c\lambda}x_d - \frac{a}{c\lambda}x_0 + \frac{b}{c\lambda}y_d - \frac{b}{c\lambda}y_0 + \frac{1}{\lambda} = 0$$

$$\implies x_d^2 + y_d^2 + \left(\frac{a}{c\lambda} - 2x_0\right)x_d + \left(\frac{b}{c\lambda} - 2y_0\right) + x_0^2 + y_0^2 - \frac{a}{c\lambda}x_0 - \frac{b}{c\lambda}y_0 + \frac{1}{\lambda} = 0$$

$$x_d^2 + y_d^2 + ex_d + fy_d + g = 0,$$

where

$$\begin{aligned} e &= \frac{a}{c\lambda} - 2x_0 \\ f &= \frac{b}{c\lambda} - 2y_0 \\ g &= x_0^2 + y_0^2 - \frac{a}{c\lambda}x_0 - \frac{b}{c\lambda}y_0 + \frac{1}{\lambda}. \end{aligned} \quad (\text{Equation 3.8})$$

It is also possible to come to the conclusion that straight lines are imaged as circles using the parametric form of a straight line (Strand & Hayman, 2005).

Following is the relationship among e , f , and g :

$$\begin{aligned} ex_0 + fy_0 + g &= \left(\frac{a}{c\lambda} - 2x_0\right)x_0 + \left(\frac{b}{c\lambda} - 2y_0\right)y_0 + x_0^2 + y_0^2 - \frac{a}{c\lambda}x_0 - \frac{b}{c\lambda}y_0 + \frac{1}{\lambda} \\ \implies x_0^2 + y_0^2 + ex_0 + fy_0 + g - \frac{1}{\lambda} &= 0 \end{aligned} \quad (\text{Equation 3.9})$$

3.1.3 Inverse mapping

When undistorting an image, it is necessary to compute, for each pixel in the output undistorted image, the corresponding pixel position in the distorted image then perform interpolation to determine the actual pixel color or intensity in the output undistorted image (we use simple bilinear interpolation in all of the experiments reported on in this paper). However, while every distorted image point (x_d, y_d) is mapped to a unique undistorted image point (x_u, y_u) by Equation 3.5, the reverse is not true. To invert Equation 3.5 and find the value of x_d and y_d as a function of x_u and y_u , we first square and add the individual equations to obtain

$$(x_u - x_0)^2 + (y_u - y_0)^2 = \frac{1}{(1 + \lambda r_d^2)^2} ((x_d - x_0)^2 + (y_d - y_0)^2). \quad (\text{Equation 3.10})$$

We then let r_u be the distance of (x_u, y_u) to the distortion center:

$$r_u^2 = (x_u - x_0)^2 + (y_u - y_0)^2.$$

This lets us simplify Equation 3.10 to

$$\implies r_d^2 = (1 + \lambda r_d^2)^2 r_u^2.$$

$$\implies \frac{r_d^2}{r_u^2} = (1 + \lambda r_d^2)^2.$$

$$\implies \frac{r_d}{r_u} = 1 + \lambda r_d^2.$$

$$\implies \lambda r_u r_d^2 - r_d + r_u = 0.$$

$$\implies r_d^2 - \frac{1}{\lambda r_u} r_d + \frac{1}{\lambda} = 0. \quad (\text{Equation 3.11})$$

For positive λ (pincushion distortion), given $0 < r_u^2 < \frac{1}{4\lambda}$, Equation 3.11 has two positive real roots. We use the smaller of the two. For negative λ (barrel distortion), given any $r_u^2 > 0$, there are always two real solutions, but one is negative. We use the positive solution. After solving for r_d in terms of r_u , the distorted image coordinates corresponding to (x_u, y_u) can be obtained as

$$\begin{aligned} x_d &= x_0 + \left(\frac{r_d}{r_u} \right) (x_u - x_0) \\ y_d &= y_0 + \left(\frac{r_d}{r_u} \right) (y_u - y_0). \end{aligned} \quad (\text{Equation 3.12})$$

3.1.4 Estimating distortion parameters from circular arcs

Strand and Hayman (2005) and Wang et al. (2009) show that it is possible to estimate λ from the parameters of circular arcs identified in an image. Wang et al. (2009) further show how both λ and the distortion center (if not assumed to be the center of the image) can be estimated from the parameters of three circular arcs identified in an image. We use their formulation. In Equation 3.8, multiplying the equation for e by x_0 , the equation for f by y_0 , and adding the equations for ex_0 , fy_0 , and g , we obtain

$$x_0^2 + y_0^2 + ex_0 + fy_0 + g - \frac{1}{\lambda} = 0 \quad (\text{Equation 3.13})$$

For each of the three arcs $i \in \{1, 2, 3\}$, we use Equation 3.13 to obtain coefficients e_i , f_i , and g_i ,

$$x_0^2 + y_0^2 + e_1 x_0 + f_1 y_0 + g_1 - \frac{1}{\lambda} = 0 \quad (\text{Equation 3.14})$$

$$x_0^2 + y_0^2 + e_2 x_0 + f_2 y_0 + g_2 - \frac{1}{\lambda} = 0 \quad (\text{Equation 3.15})$$

$$x_0^2 + y_0^2 + e_3 x_0 + f_3 y_0 + g_3 - \frac{1}{\lambda} = 0 \quad (\text{Equation 3.16})$$

subtracting Equation 3.14 and Equation 3.15.

$$(e_1 - e_2)x_0 + (f_1 - f_2)y_0 + (g_1 - g_2) = 0$$

subtracting Equation 3.14 and Equation 3.16.

$$(e_1 - e_3)x_0 + (f_1 - f_3)y_0 + (g_1 - g_3) = 0$$

then the distortion center can be estimated by solving the linear system

$$\begin{aligned}(e_1 - e_2)x_0 + (f_1 - f_2)y_0 + (g_1 - g_2) &= 0 \\ (e_1 - e_3)x_0 + (f_1 - f_3)y_0 + (g_1 - g_3) &= 0,\end{aligned}\tag{Equation 3.17}$$

and an estimate of λ can be obtained using Equation 3.9

$$\frac{1}{\lambda} = x_0^2 + y_0^2 + ex_0 + fy_0 + g\tag{Equation 3.18}$$

using any of the three arcs' parameters in place of e , f , and g . See Wang et al. (Wang et al., 2009) for details.

3.2 Robust Radial Distortion Estimation

In this section, we provide a detailed algorithm for estimating the parameters of the mathematical model introduced in Section 3.1.

3.2.1 Identifying circular arcs

The first step in our method is to robustly identify as many non-overlapping circular arcs as possible in the distorted input image. Each arc is identified by a circle center, circle radius, and the contiguous sequence of pixels consistent with that circle.

To find arcs, we first extract edges and link adjacent edge pixels remaining contours. We discard any contour whose length is less than l^{\min} pixels (we use $l^{\min} = 10$ pixels) and then we attempt to find long pixel subsequences within each contour that can be fit by circular arcs. Our method is based on random sampling and inspired by RANSAC (Fischler & Bolles, 1981), but, rather than finding a single model for all the data, we preserve all models (candidate circular arcs) not overlapping with other arcs in the same contour that have more support. The termination criterion is to stop once the probability that an arc of minimal length has not yet been found is small.

In Algorithm 2, we provide the details of the method.¹ To determine the number of sampling iterations required, the algorithm uses a function $f(l, n)$, which gives the number of trials required to ensure that the probability of not sampling three of l inliers from a set of n points is small. This ensures that we sample a sufficient number of times to find, with high probability, all arcs with sufficient length in each contour.

¹Algorithm 2 appeared as Algorithm 1 in (Bukhari & Dailey, 2010) and (Bukhari & Dailey, 2012).

```

Input: Contours  $C_1, C_2, \dots$ 
Output:  $A$  is the output arc set
 $A \leftarrow \emptyset$ 
foreach contour  $C_i$  do
    if  $|C_i| \geq l^{\min}$  then
         $N \leftarrow f(l^{\min}, |C_i|)$ 
        for  $n \leftarrow 1$  to  $N$  do
            Sample three points  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  from  $C_i$ .
            if  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  are not collinear then
                Calculate  $x_c, y_c, r$  from  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 
                 $A^{\text{new}} \leftarrow$  arc for longest subsequence of  $C_i$  consistent with  $x_c, y_c, r$ 
                if  $|A^{\text{new}}| \geq l^{\min}$  then
                    if  $A^{\text{new}}$  does not overlap with any arc in  $A$  then
                         $A \leftarrow A \cup \{A^{\text{new}}\}$ 
                    end
                    else if  $A^{\text{new}}$  is longer than every overlapping arc in  $A$  then
                        Remove arcs overlapping with  $A^{\text{new}}$  from  $A$ 
                         $A \leftarrow A \cup \{A^{\text{new}}\}$ 
                    end
                end
            end
        end
    end
end

```

Algorithm 2: Robust arc identification.

3.2.2 Refining circular arc estimates

After the initial arc identification process is complete, each resulting arc, whose parameters have been calculated directly from the minimum sample of three points, is refined using the inlier pixel contour subsegment supporting that model. The gold standard algorithm is discussed in section 2.10.1. To obtain the initial guess, we use a variety of methods as detailed in the next section.

3.2.3 Results of third stage refinements

We defined a third stage refinements as a global LM type optimization procedure (taking into account all inlier arcs) to refine parameters by minimizing the distance of the distortion corrected arc points to the their associated best line. Previously as explained, we did use a least squares procedure based an all inlier arcs to reestimate the distortion parameters. However, this is an algebraic minimization, not a geometric minimization, and it did not take into account the possible errors in the arc estimates. Since it is useful to compare this algebraic method with a full LM type algorithm having a geometric objective function based on how close each contour pixel is to the orthogonal regression line after undistorting, we implemented the suggested third stage optimization, and repeated some of the experiments with the suggested optimization. A typical runtime for this further optimization step alone

is 9.23661s. It only slightly reduced the sum of squares of residuals of projected undistorted points from the undistorted points. Since our residuals are already very small (for example a sum squared error of 61.1524 over 92 inlier arcs consisting of 29520 pixels), and since we do not find any substantial difference between the residuals after refining the parameters with the global LM, (17 iterations of LM reduced the residual to 59.3572 in 9.23661s), we believe the improvement is too small to justify the additional CPU time.

3.2.4 Algebraic circle fitting methods

As the initial estimate of the circle's parameters, we use either the parameters calculated during the sampling procedure or one of three circle fitting methods, Pratt (1987), Taubin (1991), and Kukush-Markovsky-van-Huffel (KMvH) (2004), based on algebraic error minimization. See 2.9 for details about each algebraic circle fitting methods.

3.2.5 Geometric circle fitting methods

Researchers have proposed several circle-specific parameter refinement methods based on LM. In addition to generic LM, in this paper, we also experiment with Trust-Region-LM (1978), Chernov and Lesort LM (Chernov-LM) (2005), and Reduced-LM (2010). See 2.10 for details about each geometric circle fitting methods.

Also refer to Table 2.1 for a comparative summary of the different circle fitting algorithms.

We coded all the geometric and algebraic circle fitting methods in C++ using OpenCV (Bradski, 2000), with reference to Chernov's (1997) MATLAB implementations.

3.2.6 Experimental design for circular arc estimation methods

We have found that different circle fitting methods provide very different performance in terms of radial distortion estimation. In the experimental evaluation (Section 4.1), we therefore perform a comprehensive study of the effect the circle fitting algorithm has on radial distortion correction. We use 10 different variations of the fitting methods described in the previous section. For a comparative summary of the different circle fitting algorithms, refer to Table 2.1, and for more detailed discussion of the methods, refer to Chernov (Chernov, 2010).

In our experiments, the “Ransac” method means we simply accept the circular arc model computed from three sample points, without any refinement after calculating inliers. “Ransac-Taubin,” “Ransac-Pratt,” and “Ransac-KMvH” are the results of using the Taubin, Pratt, or KMvH methods to refine the arc models computed from three sample points. The names “Ransac-LM,” “Ransac-TR-LM,” “Ransac-Chernov-LM,” and “Ransac-Reduced-LM” denote the application of general LM, Trust Region LM, Chernov's LM, or the Reduced LM methods previously described directly to the models computed from three sample points. Under the hypothesis that starting LM from the sample-based estimate might not work as well as an initial estimate closer to the optimum, we also performed two series of experiments

in which we first applied either the Taubin or the Pratt methods to the sample-based models then applied general LM to the Taubin or the Pratt estimates respectively. The results from these methods are shown as “Ransac-Taubin-LM” and “Ransac-Pratt-LM.”

3.2.7 Estimating distortion parameters

Once we have obtained a set of circular arcs as candidate distorted straight lines, we use the estimator of Equation 3.17 and Equation 3.18 and a RANSAC procedure to find the set of distortion parameters with maximal support. However, we use a modified notion of support; rather than counting the number of arcs fit by a particular model, we count the sum of the lengths (in pixels) of the arcs. Longer arcs provide much more accurate parameter estimates than shorter arcs. The weighted support strategy emphasizes models that fit as many long arcs as possible rather than models that fit many small arcs. In the sampling loop, we sample three arcs, calculate the model, then classify each arc as an inlier if, after undistortion, the arc’s pixels fit a straight line.

In Algorithm 3, we provide the details of the method ². In the sampling loop, we use adaptive calculation of the number of iterations required based on the highest number of pixels in inlier arcs seen so far. The termination criterion uses the same function $f(l, n)$ previously introduced to determine the number of trials required to ensure that the probability of never sampling three arcs from a consensus set of size (in pixels) at least the size of the current support set is small.

To judge whether an arc is an inlier for distortion parameters (λ, x_0, y_0) , we first perform orthogonal regression to determine the line best fitting the pixels of the arc before and after undistortion using the candidate parameters. We use two criteria: the undistorted arc should be very close to the straight line in terms of RMSE, and the RMSE for the undistorted arc should be smaller than that of the original distorted arc. The second criterion is necessary to handle arcs that are already very close to straight lines in the original distorted image. We require that these arcs should not only be nearly straight after undistortion, but that they should also be more straight than they were before undistortion. This avoids situations in which incorrect distortion parameters make nearly straight contours less straight but still close enough to straight to be counted as inliers.

Once the required number of iterations have been performed, we obtain a final least squares estimate of the distortion parameters based on all inlier arcs and Equation 3.17 and Equation 3.18. From each possible pair of inlier arcs, we form the linear system described by Equation 3.17 and find the least-squares solution for the distortion center (x_0, y_0) . After estimating the distortion center, we estimate λ using Equation 3.18 and all inlier arcs.

²Algorithm 3 appeared as Algorithm 2 in Bukhari and Dailey (2010) and Bukhari and Dailey (2012). Initial version of Algorithm 3 appeared in Bukhari and Dailey (2010) and after some modification in the Algorithm 3, the final version of the algorithm appeared in Bukhari and Dailey (2010)

```

Input: Arc set  $A$ 
Output:  $\lambda^*, x_0^*, y_0^*$  are the output distortion parameters
begin
     $(\lambda^*, x_0^*, y_0^*) \leftarrow (\emptyset, \emptyset, \emptyset)$ 
    if  $|A| \geq 3$  then
         $n \leftarrow 0$ 
         $s^* \leftarrow 0$ 
        while true do
             $n \leftarrow n + 1$ 
            Sample three distinct arcs  $A_1, A_2, A_3$  with probability proportional to lengths  $|A_i|$ .
            Estimate  $(\lambda, x_0, y_0)$  from  $A_1, A_2, A_3$  per Equation 3.17 and Equation 3.18
             $s \leftarrow$  support (in pixels) for  $(\lambda, x_0, y_0)$ 
            if  $s > s^*$  then
                 $s^* \leftarrow s$ 
                 $(\lambda^*, x_0^*, y_0^*) \leftarrow (\lambda, x_0, y_0)$ 
            end
            if  $n \geq f(s, \sum_{i=1}^{|A|} |A_i|)$  then
                break
            end
        end
        Reestimate  $(\lambda^*, x_0^*, y_0^*)$  using inlier arcs.
    end
end

```

Algorithm 3: Robust distortion parameter estimation.

3.2.8 Undistortion

The last step in our procedure is to undistort the input image. We use the optimal distortion parameters computed per the previous section and the inverse of the distortion model in Equation 3.12 with bilinear interpolation and appropriate translation and scale factors to produce the output undistorted image.

Chapter 4

Experiments and Results

This chapter describes the experiments and the results of this dissertation. It explains experiments on synthetic and real images. It provides detail of statistical analysis and a comparison with the Alvarez et al. (2009) method. The chapter is divided in the following sections: Quantitative Results on Synthetic Images, Qualitative Results on Real Images, and Comparison with Alvarez et al. (2009) method.

4.1 Quantitative Results on Synthetic Images

In this section, we describe a detailed quantitative study of the performance of our method on synthetic images. A sample of the synthetic images we use with results is shown in Figure 4.1. We used the same original image (Figure 4.1(a)) for all synthetic image experiments. In each experiment, we distort the original image using particular ground truth values for λ , x_0 , and y_0 (Figure 4.1(b)), extract Canny edges (Figure 4.1(c)), link the edge pixels into contours (Figure 4.1(d)), identify circular arcs among the contours (Figure 4.1(e)), estimate the distortion parameters, and then use those parameters to undistort the image.

To evaluate the quality of reconstruction of the original synthetic image, we use root mean-squared error (RMSE) and peak signal-to-noise ratio (PNSR), the most commonly used image quality metrics for cases in which the original undistorted image is available for comparison.

4.1.1 Synthetic images

We performed two series of experiments with synthetic images. For edge extraction, we modified OpenCV’s Canny edge detector to automatically select a low gradient threshold and a high gradient threshold based on a cumulative histogram of magnitudes, similar to the Matlab implementation of Canny’s method (MATLAB, 2009).

4.1.1.1 Experiment 1 (varying λ)

In a first series of runs, we varied λ while keeping the distortion center fixed at $(x_0, y_0) = (320, 240)$ but estimated all three parameters. For each level of λ , we compare 10 methods for circular arc estimation.

Figure 4.2 and Figure 4.3 show the distorted image and a sample undistorted result for each level of λ using the “Ransac-LM” method. For the extreme case of $\lambda = 10^{-5}$, undistorted image points map to multiple valid distorted image points, so we only map the points for which $r_d^2 < \lambda$, resulting in a circular valid region around the image center.

To precisely quantify the performance of each algorithm, for each level of λ and each circle fitting method, we performed 10 runs with different random seeds and collected three measurements in each case: the absolute relative estimation error for λ , i.e., $|(\lambda_{\text{est}} - \lambda_{\text{true}})/\lambda_{\text{true}}|$, RMSE, and PNSR. For the extreme case of $\lambda = 10^{-5}$, we only calculated RMSE and PNSR over the circular valid region shown in Figure 4.3. The mean measurements with 95% confidence intervals are shown in Figure 4.7(a–c).

The results in terms of relative estimation error for λ in Figure 4.7(a) show quite clearly that our method is extremely accurate at estimating λ for moderate ($\lambda = \pm 10^{-6}$) or extreme ($\lambda = \pm 10^{-5}$) distortion but quite inaccurate for very small levels of distortion. The inaccuracy with small distortion levels reflects two factors. First, since the ground truth value is extremely small in the first place, small deviations between estimated and ground truth parameter values give large relative errors. Second, when our algorithm fails to find a sufficient number of contours that can be modeled as circular arcs, it defaults to an estimate of $\lambda = 0$, which leads to a relative error of 1. Fortunately, as the RMSE and PNSR comparisons show, this relative inaccuracy for small λ does not affect our method’s ability to reconstruct the original undistorted image.

Since the RMSE and PNSR results shown in Figure 4.7(b–c) are difficult to interpret, we performed a series of statistical analyses. As the dependent variables, we used pixel intensity RMSE and PNSR. As the independent variables, we used the algorithm and the different levels of λ .

For both RMSE and PNSR, a two-way analysis of variance (ANOVA) revealed main effects for both predictor variables and an interaction. To understand the main effect of differing ability of each algorithm to reconstruct the original image, we performed a post-hoc analysis using the Tukey correction for all pairwise comparisons among the 10 different algorithms.

For both RMSE and PNSR, the best numerical results were obtained with Ransac-Pratt, but statistically, according to both measures, four algorithms, namely Ransac-Pratt, Ransac-Pratt-LM, Ransac-LM, Ransac-Reduced-LM, Ransac-Taubin, and Ransac-Taubin-LM are equivalent to and better than the remaining four algorithms (we use a familywise $\alpha = 0.05$ for all significance tests). Next are Ransac-KMvH, Ransac-Chernov-LM, and Ransac-TR-LM. These three algorithms are statistically equivalent. The last method, Ransac, is statistically worse than all the other algorithms.

4.1.1.2 Experiment 2 (varying distortion center)

In a second series of runs, we kept the distortion fixed at a moderate level of barrel distortion ($\lambda = -10^{-6}$) while varying the distortion center, then we estimated all three parameters of the distortion model. We used the same 10 circle fitting methods as in the first series of experiments manipulating λ . For each run, we measured the Euclidean distance between the estimated and ground truth distortion center as well as RMSE and PNSR. The results are shown in Figure 4.7(d–f).

To further understand the results, we performed statistical analyses with all three dependent measures. As the independent variables, we used the algorithm and the different levels of distortion center. To simplify the presentation, we name the distortion center levels as $DC_1 = 0.0$ (distortion center (320, 240)), $DC_2 = 14.1$ (distortion center (330, 250)), DC_3

$= 28.3$ (distortion center (340, 260)), $DC_4 = 42.4$ (distortion center (350, 270)), $DC_5 = 56.6$ (distortion center (360, 280)), $DC_6 = 70.7$ (distortion center (370, 290)), $DC_7 = 84.9$ (distortion center (380, 300)), $DC_8 = 99.0$ (distortion center (390, 310)).

For all three dependent measures, two-way analyses of variance (ANOVAs) revealed main effects for both predictor variables and an interaction. We then performed post-hoc Tukey comparisons among the different algorithms and among the different distortion center distances.

For distortion center estimation error, Ransac-LM was numerically best but Ransac-LM, Ransac-Reduced-LM, Ransac-Pratt, Ransac-Tauber-LM, Ransac-KMvH, and Ransac-Pratt-LM were statistically equivalent and better than the other four algorithms. Distortion center estimation error was also affected by the distance between the distortion center from the image center. The results for DC_1 were significantly better than those for DC_2 , and DC_7 was better than DC_8 . However, DC_6 happened to be as easy as DC_4 and easier than either DC_5 or DC_7 . Inspection of Figure 4.7(d) indicates that the increasing trend is driven by just two or three algorithms.

For pixel intensity RMSE, Ransac-LM was numerically best but statistically equivalent to Ransac-Pratt. The next set was Ransac-Reduced-LM, Ransac-Tauber-LM, Ransac-Pratt-LM, Ransac-KMvH, and Ransac-Tauber. Statistically, these algorithms were equivalent to each other and better than the remaining three algorithms. There was also an effect of distortion center distance on RMSE; the results for DC_1 , DC_2 , DC_3 , and DC_4 were statistically equivalent but the results for DC_6 , DC_7 , DC_4 , and DC_8 were all significantly different with respectively increasing RMSE.

Finally, the ordering in terms of increasing PNSR was Ransac-LM, Ransac-Pratt, Ransac-Reduced-LM, Ransac-Tauber-LM, Ransac-Pratt-LM, Ransac-KMvH, Ransac-Tauber, Ransac-TR-LM, Ransac, Ransac-Chernov-LM. All differences were significant except among Ransac-Pratt-LM, Ransac-KM-vH, and Ransac-Tauber, which were statistically equivalent. Distortion center distance also had some effect, with DC_1 , DC_2 , and DC_3 yielding the best PNSR. These levels were significantly better than the others, for which the trend was DC_6 , DC_4 , DC_7 , DC_5 , DC_8 , with significantly increasing PNSR, respectively.

4.1.1.3 Discussion of synthetic image experiments

Over the two series of runs, we observe that with moderate or extreme distortion, our method readily identifies the parameters of the distortion model and is successful at reconstructing the original undistorted image with high accuracy. For lower levels of distortion, the model parameters are more difficult to estimate accurately, but this inaccuracy does not affect the reconstruction results by much: Ransac-LM introduces at most only 2.5 times the reconstruction error of bilinear interpolation with the ground truth parameters.

Although some sensitivity to distortion center distance is observed, we can see readily from the data that this is only for some algorithms. The best algorithms such as Ransac-LM and Ransac-Pratt are clearly not affected by this factor. This is an improvement over other work (Kukelova & Pajdla, 2011; Alvarez et al., 2009; Brauer-Burchardt & Voss, 2001; Strand & Hayman, 2005; Fitzgibbon, 2001).

Statistically, Ransac-LM and Ransac-Pratt are the winning algorithms. It is difficult to choose between the two since they both yield excellent performance. Ransac-Pratt is more computationally efficient. Ransac-LM provides significantly better PNSR over varying distortion center distances, and it has a somewhat lower maximum RMSE over varying levels of distortion. For these reasons, we have a slight preference for Ransac-LM.

Our method selects the distortion model consistent with the largest possible number of arcs found in the image. It could certainly be fooled by a synthetic image contain a group of real world curves that happen to be consistent with each other, leading to incorrect distortion parameter estimation. However such egregious cases are extremely unlikely to arise in practice. So long as there are several distorted straight lines, our algorithm will find them and estimate a distortion model to undistort those lines ignoring the curves or outliers.

4.2 Qualitative Results on Real Images

Next, we present a qualitative evaluation of the proposed method’s ability to identify distortion parameters in several challenging real images from the Web and other papers on radial distortion estimation. The set contains images with severe barrel and pincushion distortion, showing the effects of fisheye lenses and wide angle lenses. Figure 4.8 shows step-by-step results for one of the images, from Ociepka (Ociepka, 2003). Note that some contours in Figure 4.8(c) cannot be modeled as circular arcs. The algorithm discards contours due to 1) our criteria for circular arc selection is that a contiguous sequence of pixels must be consistent with a circle, 2) the radius of an identified circular arc should be less than the five times the image width, in order to discard long straight lines in the distorted image, and 3) we discard any contour whose length is less than 10 pixels, as small arcs tend to give suboptimal estimates of the distortion parameters. Figure 4.9 summarizes the results for all of the images from the Web and previous papers on distortion estimation (Barreto & Daniilidis, 2005; Rosten & Loveland, 2011; Swaminathan & Nayar, 2000; Sarge, n.d.; Thormählen et al., 2003; Chen et al., 2009; Bockaert, n.d.; El-Melegy & Farag, 2003; Chanel, n.d.; Alvarez et al., 2010; Whittaker, n.d.; Skewes, n.d.; Solheim, n.d.; Tomasi, 2007; Laksi, n.d.; Yusuf, n.d.; Bucket, n.d.; Dyer, n.d.; Oleson, n.d.; Rideout, 2002; Kbh3rd, 2008). These results indicate the robustness and accuracy of our procedure. Many are difficult due to severe fisheye distortion and circular arcs that are not straight lines in the real world. Despite these challenges, our robust arc selection method is able to find consensus sets corresponding to distorted straight lines and is successful at removing most of the radial distortion from the images.

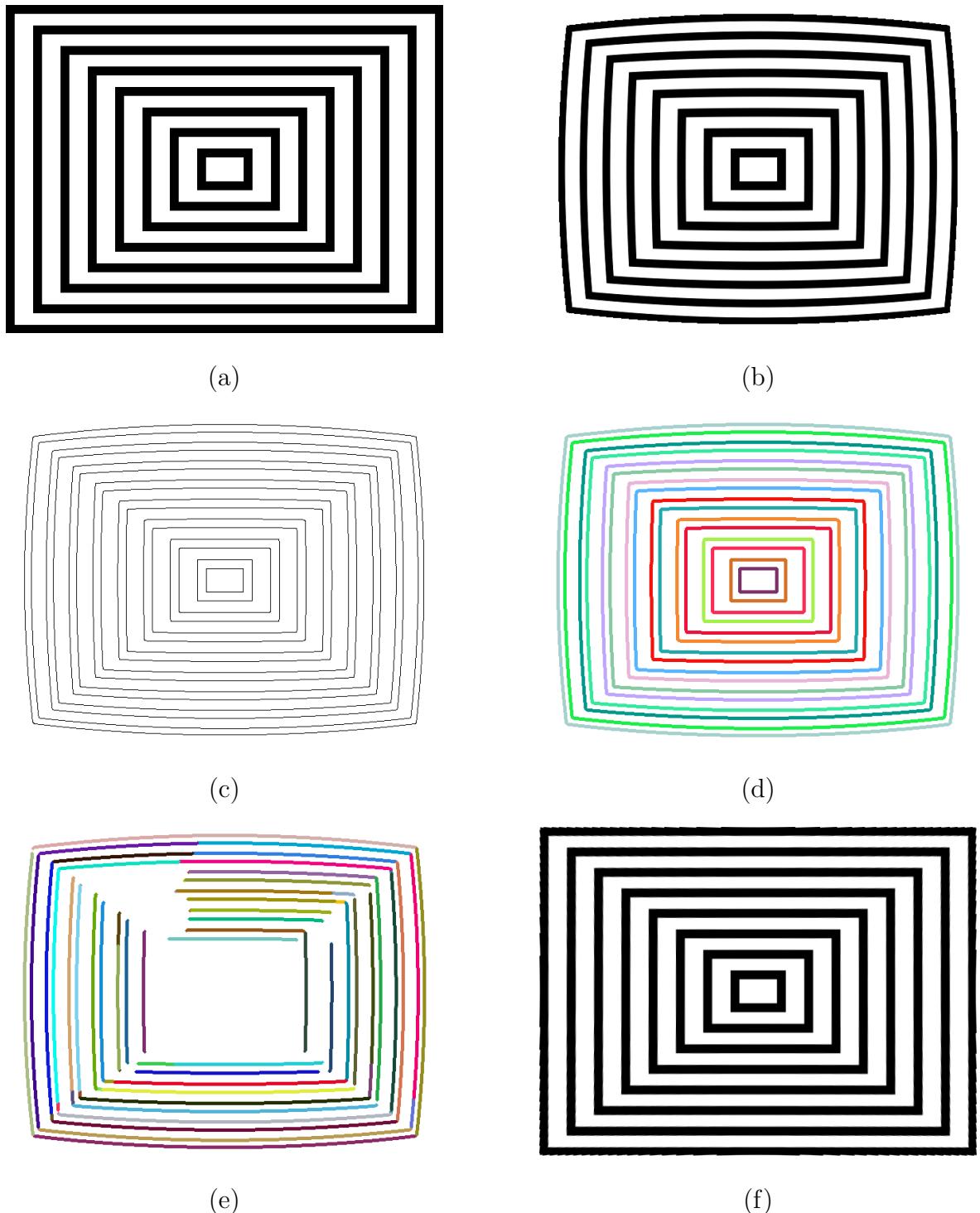
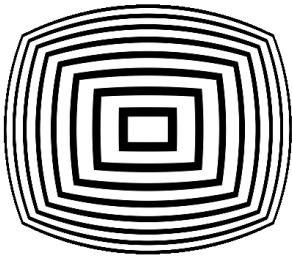
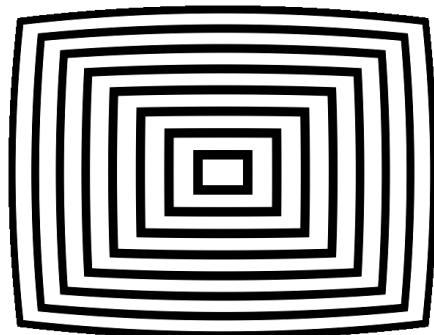


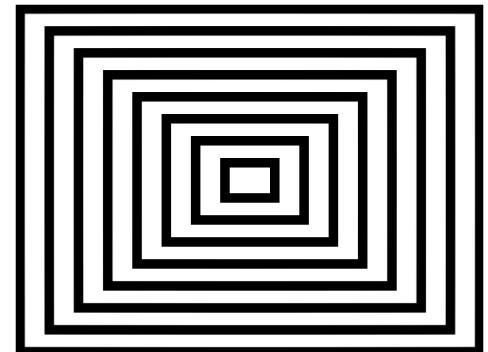
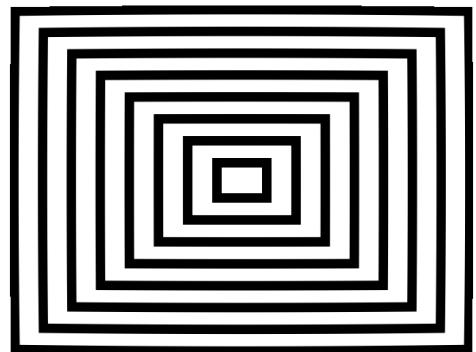
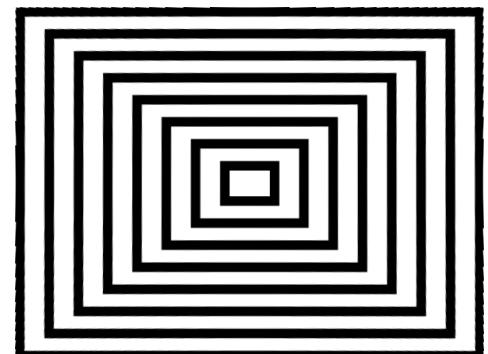
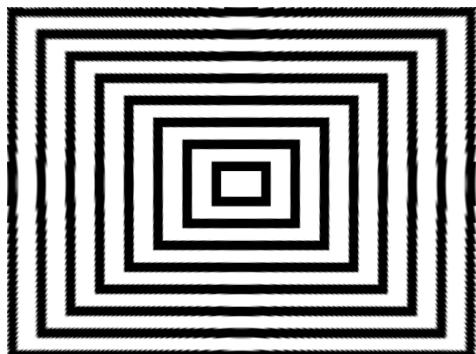
Figure 4.1: Example experiment with synthetic image size 640×480 . (a) Original image. (b) Distorted image with $\lambda = -10^{-6}$ and $(x_0, y_0) = (320, 240)$ (the image center). (c) Canny edges (d) Extracted contours. (e) Estimated arcs. (f) Undistorted image using estimated values of $\lambda = -1.00419e^{-6}$, $x_0 = 319.352$, and $y_0 = 238.009$. Using true parameters RMSE = 3.27813 and PNSR = 37.8183 dB; Using estimated parameters RMSE = 3.86511 and PNSR = 36.3876 dB.



(a) $\lambda = -10^{-5}$



(b) $\lambda = -10^{-6}$



(e) $\lambda = -10^{-7}$

(f) $\lambda = -10^{-8}$

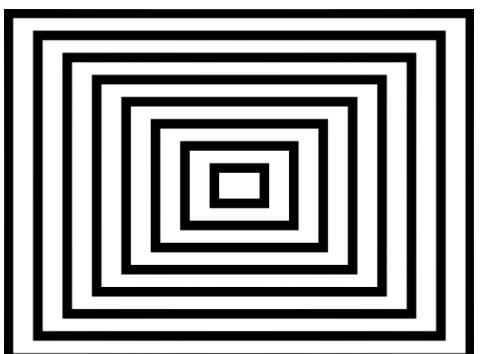
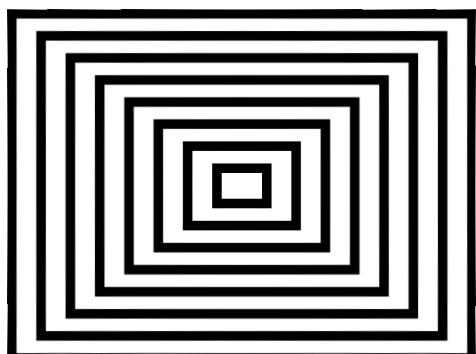
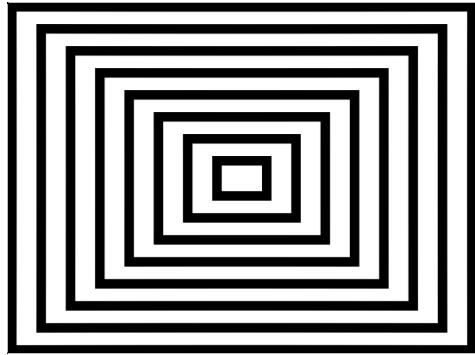
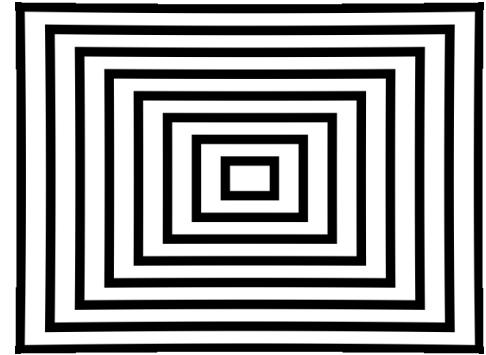


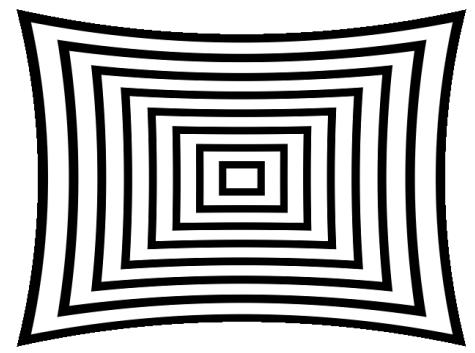
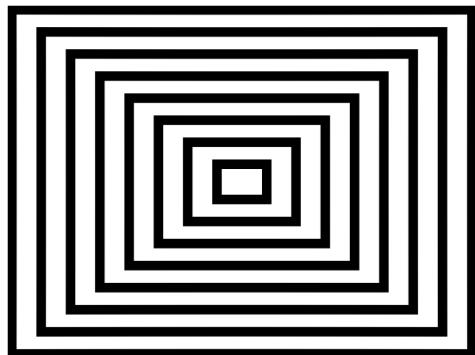
Figure 4.2: Experiment 1. Undistortion of synthetic images. Image size is 640×480 and distortion center is $(320, 240)$. First row and third row: distorted images at different levels of λ . Second row and fourth row: corresponding undistorted images using parameters estimated by the “Ransac-LM” circle fitting method.



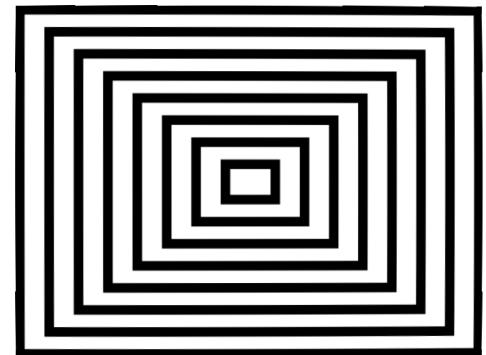
(a) $\lambda = 10^{-8}$



(b) $\lambda = 10^{-7}$



(e) $\lambda = 10^{-6}$



(f) $\lambda = 10^{-5}$

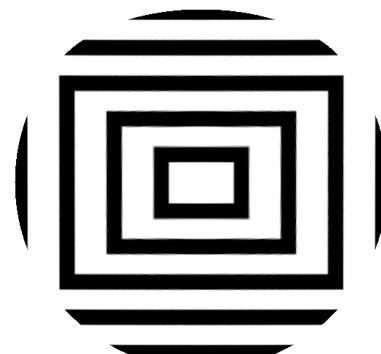
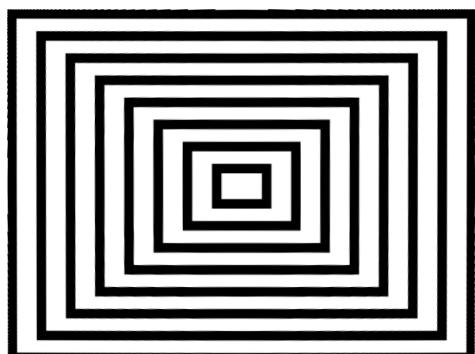
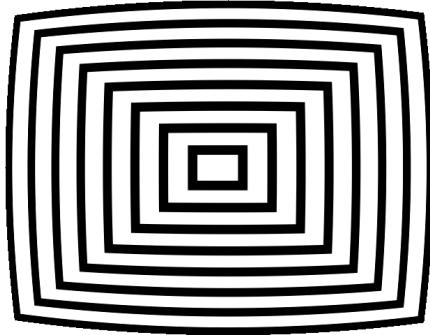
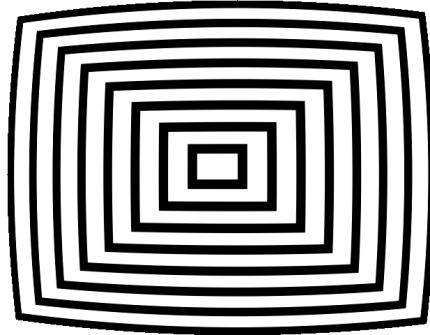


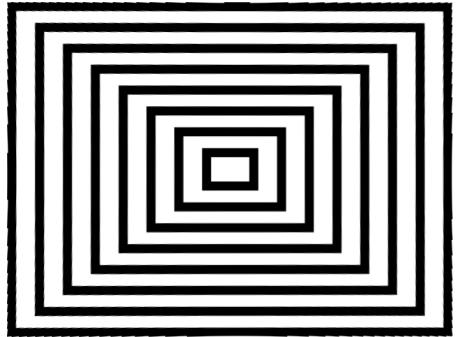
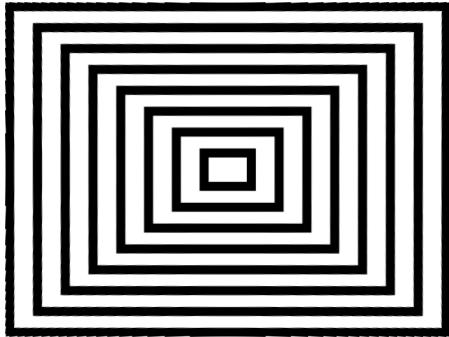
Figure 4.3: Experiment 1. Undistortion of synthetic images. Image size is 640×480 and distortion center is $(320, 240)$. First row and third row: distorted images at different levels of λ . Second row and fourth row: corresponding undistorted images using parameters estimated by the “Ransac-LM” circle fitting method.



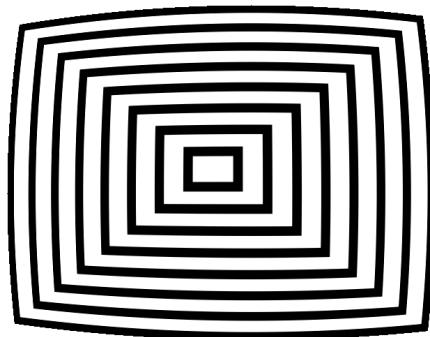
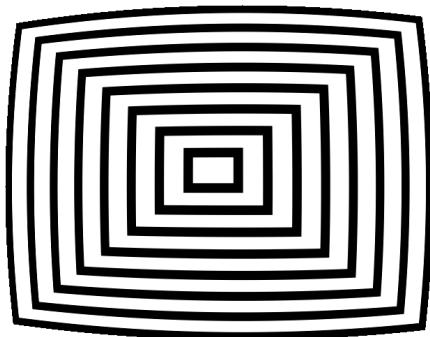
(a) $DC(320, 240)$



(b) $DC(330, 250)$



(b) $DC(330, 250)$



(e) $DC(340, 260)$

(f) $DC(350, 270)$

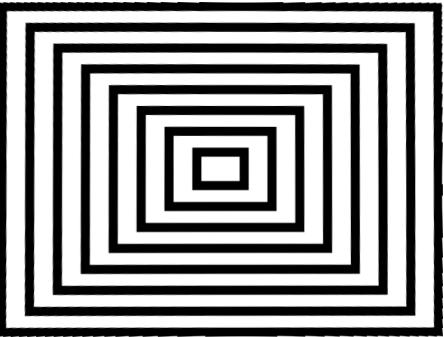
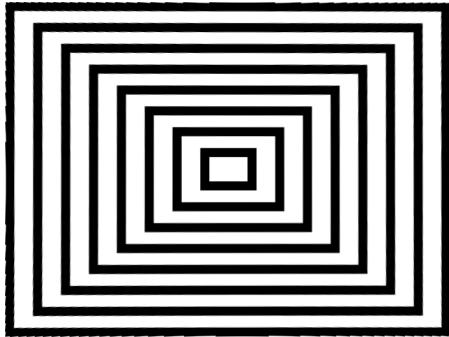
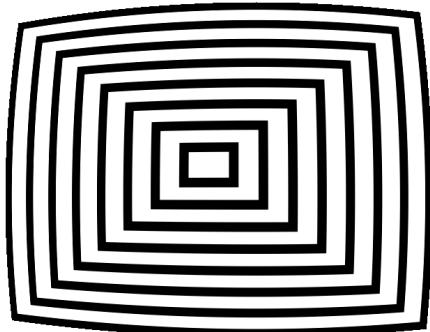
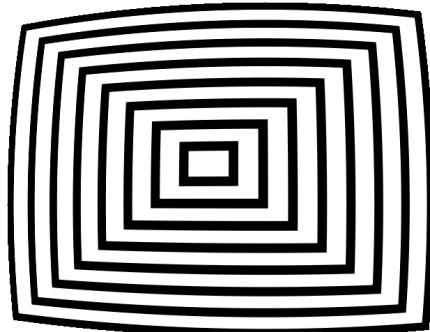


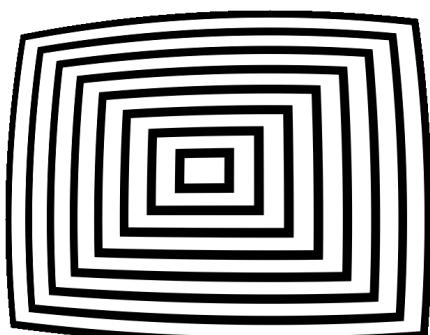
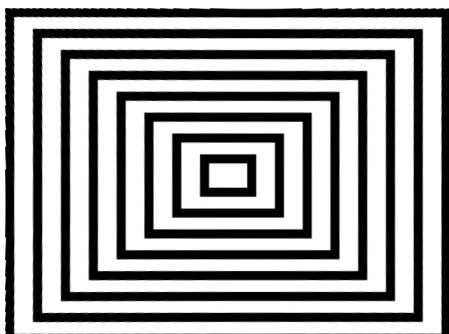
Figure 4.4: Experiment 2. Undistortion of synthetic images. Image size is 640×480 and distortion center is $(320, 240)$, $(330, 250)$, $(340, 260)$, and $(350, 270)$ respectively . First row and third row: distorted images at different levels of distortion center (DC). Second row and fourth row: corresponding undistorted images using parameters estimated by the “Ransac-LM” circle fitting method.



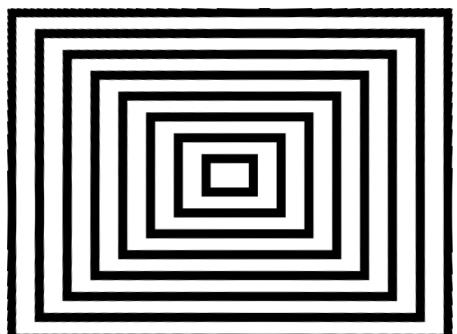
(a) $DC(360, 280)$



(b) $DC(370, 290)$



(e) $DC(380, 300)$



(f) DC (390, 310)

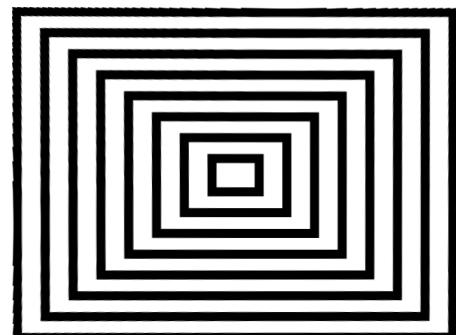
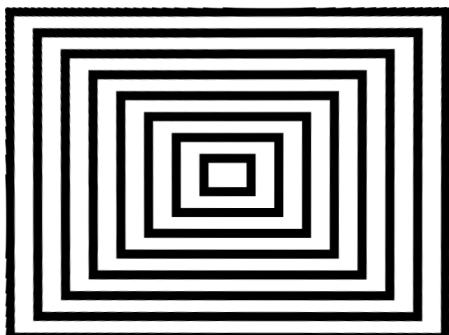


Figure 4.5: Experiment 2. Undistortion of synthetic images. Image size is 640×480 and distortion center is $(360, 280)$, $(370, 290)$, $(380, 300)$, and $(390, 310)$ respectively. First row and third row: distorted images at different levels of distortion center (DC). Second row and fourth row: corresponding undistorted images using parameters estimated by the “Ransac-LM” circle fitting method.

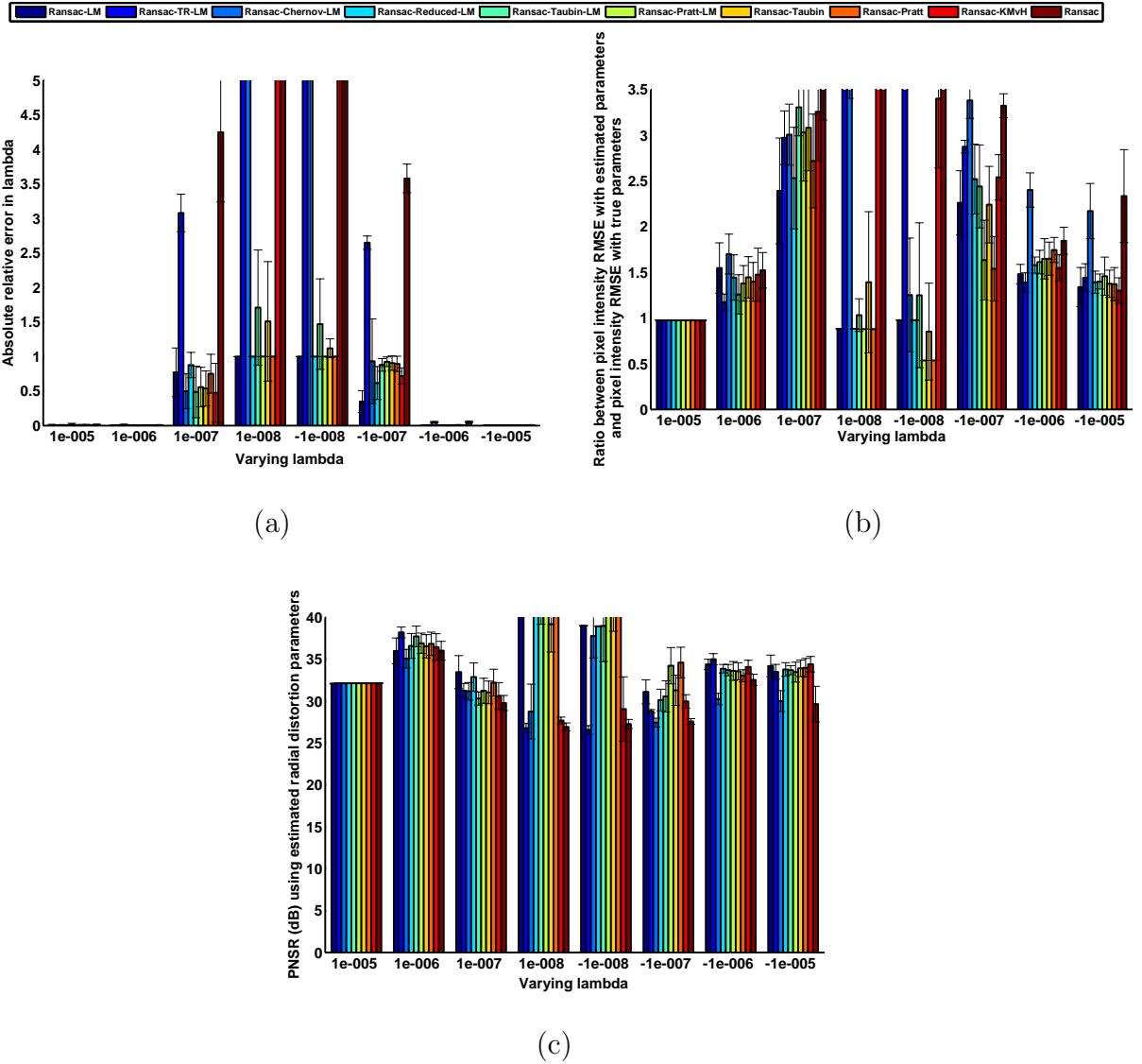


Figure 4.6: Results of experiment 1 using synthetic images. (a-c) represent results with varying λ , with distortion center fixed at the image center. All graphs show averages over 10 runs with error bars showing 95% confidence intervals on the mean. (a) True versus estimated λ . (b) Average RMSE. (c) Average PNSR.

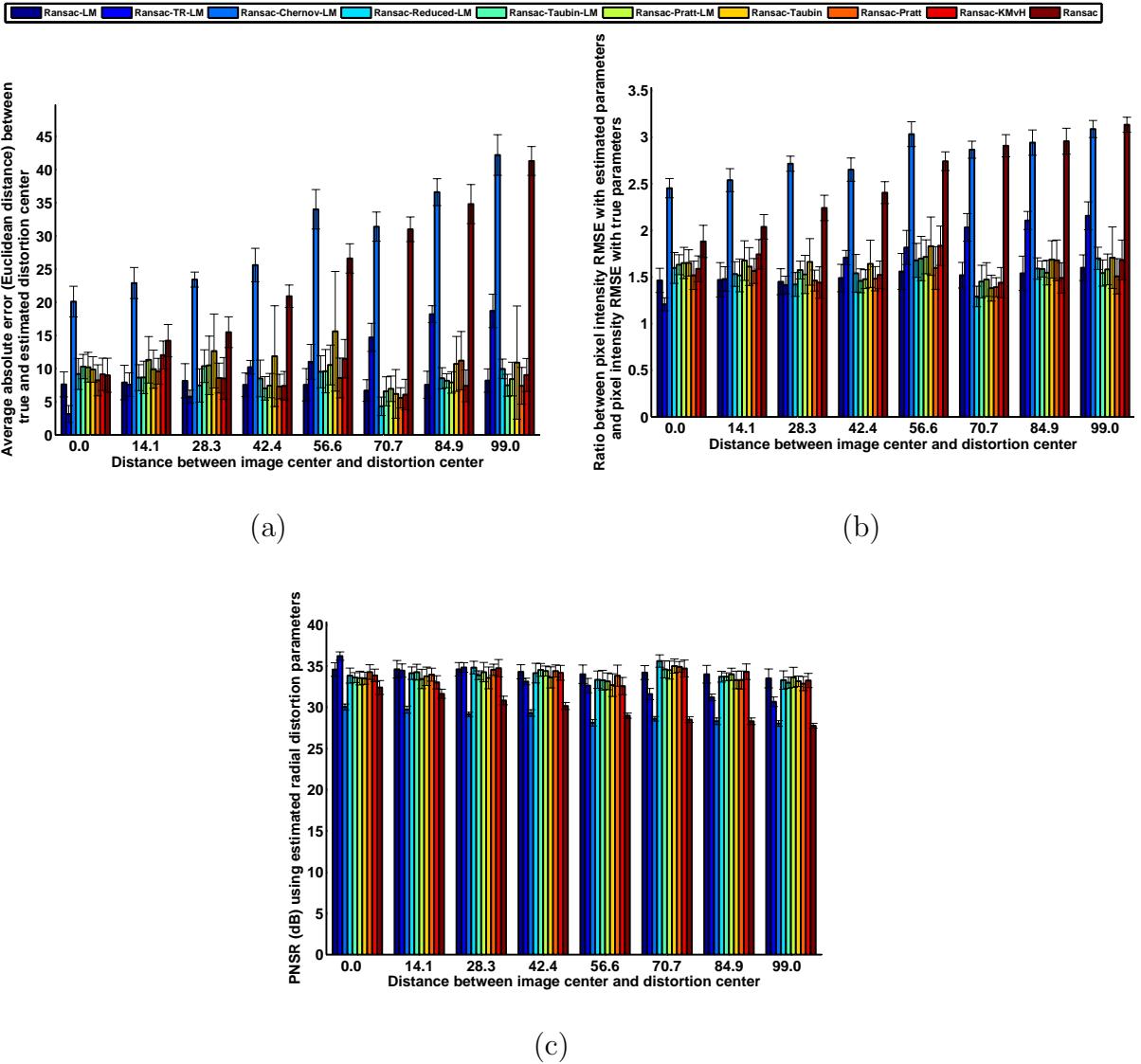


Figure 4.7: Results of experiment 2 using synthetic images. (a-c) represent results with varying distance of the distortion center to the center of the image. λ is fixed at -10^{-6} . All graphs show averages over 10 runs with error bars showing 95% confidence intervals on the mean. (a) True versus estimated distance of the distortion center from the image center. (b) Average RMSE. (c) Average PNSR.

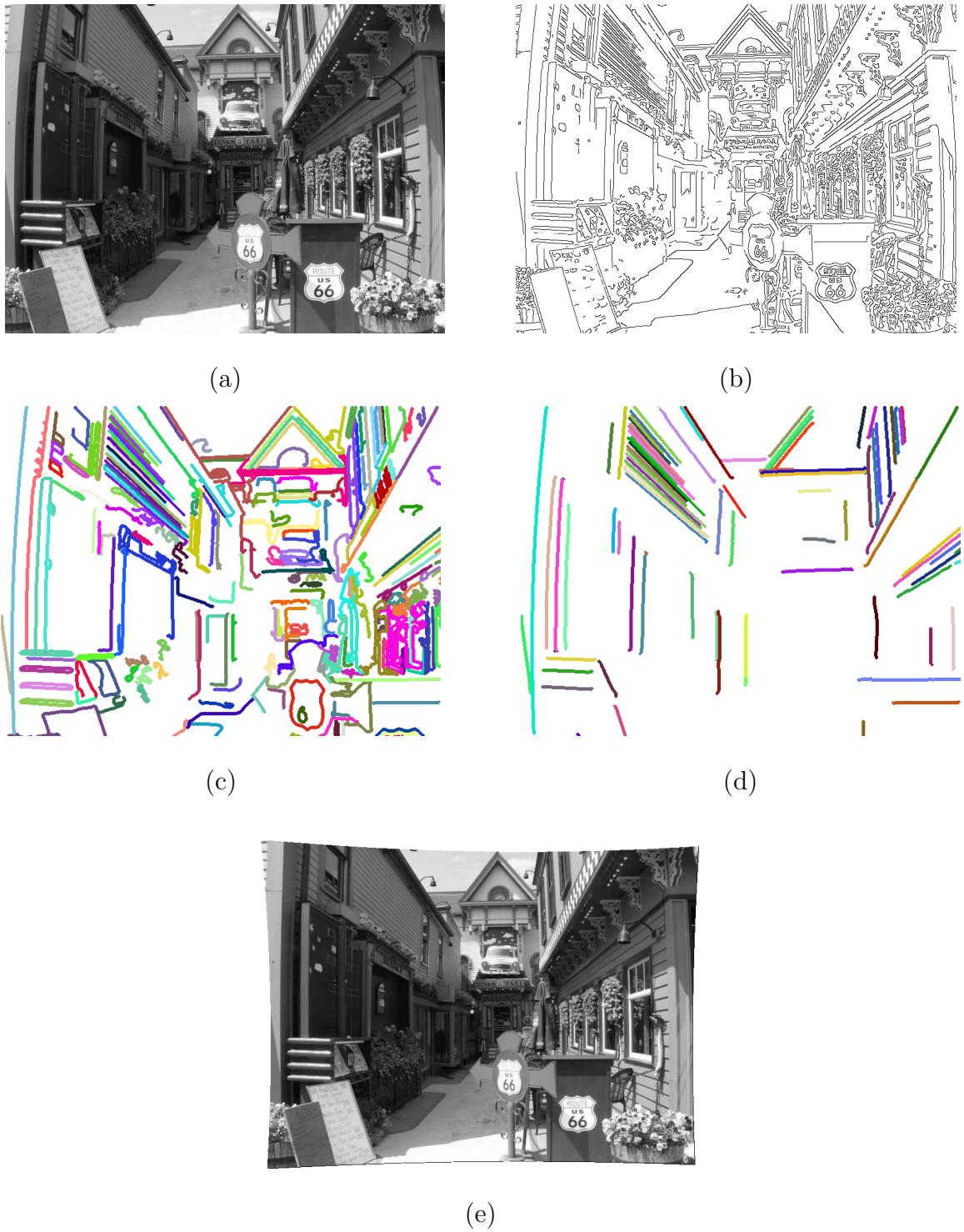


Figure 4.8: Example results on real image. (a) Original image. (b) Detected edges. (c) Extracted contours. (d) Identified arcs. (e) Undistorted image using parameters estimated via the “Ransac-LM” circle fitting method. The distorted image is taken from Ociepka Ociepka (2003).

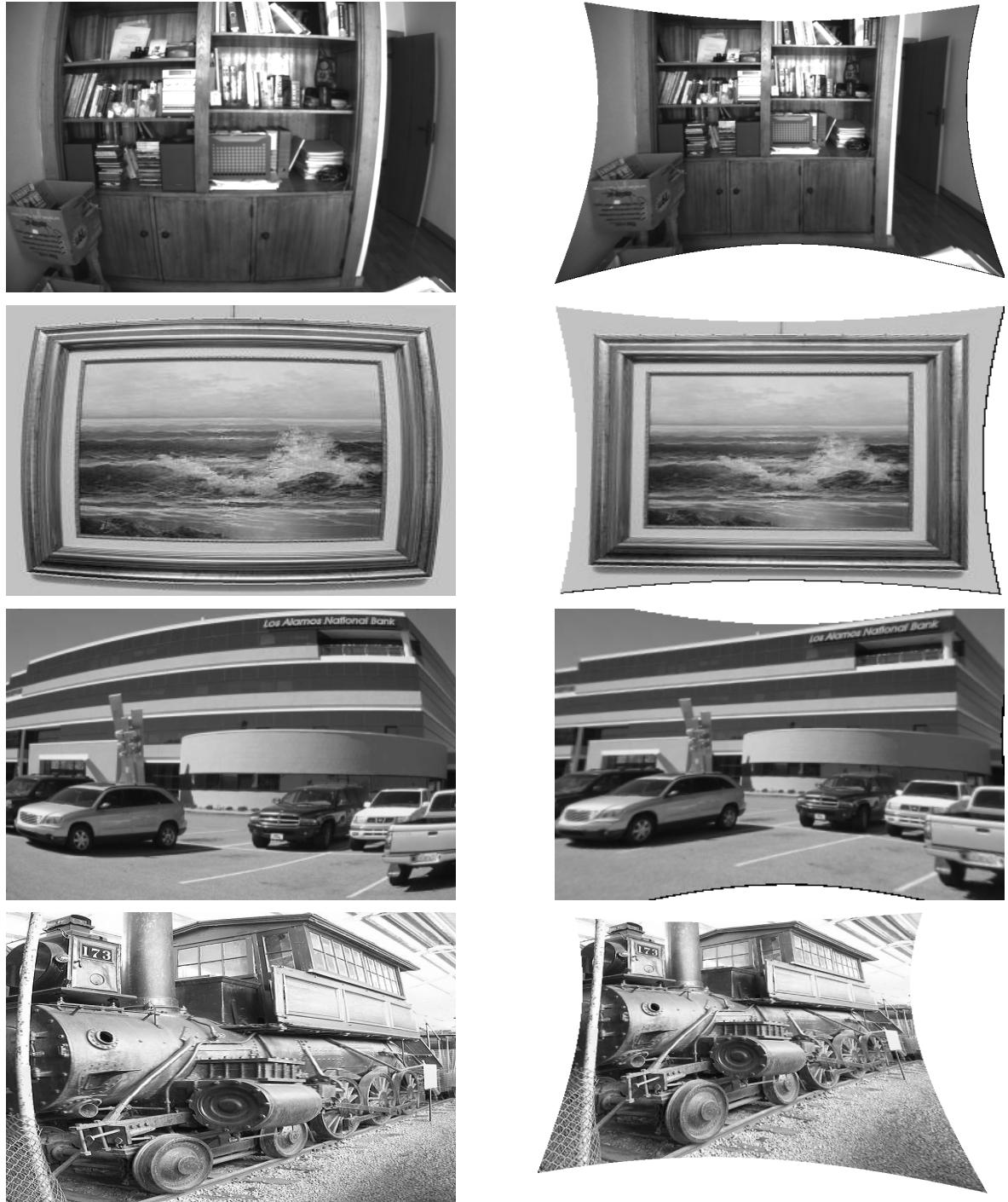
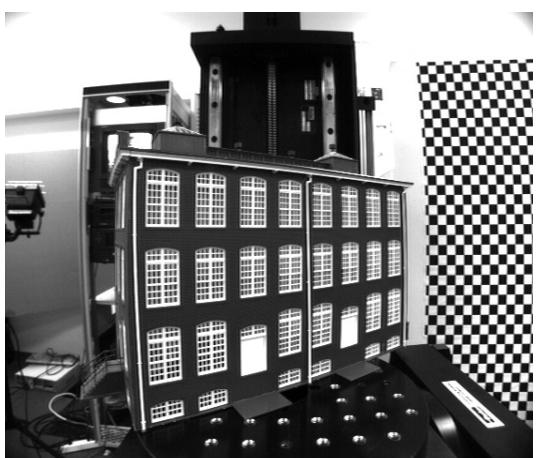


Figure 4.9: Undistortion of real images. Column 1, column 3, and column 5 represent original images. Column 2, column 4, and column 6 represent undistorted images using parameters estimated via the “Ransac-LM” circle fitting method. The distorted images are taken from several sources. Barreto and Daniilidis (2005); Rosten and Loveland (2011); Swaminathan and Nayar (2000); Sarge (n.d.); Thormählen et al. (2003); Chen et al. (2009); Bockaert (n.d.); El-Melegy and Farag (2003); Chanel (n.d.); Alvarez et al. (2010); Whittaker (n.d.); Skewes (n.d.); Solheim (n.d.); Tomasi (2007); Laksi (n.d.); Yusuf (n.d.); Bucket (n.d.); Dyer (n.d.); Oleson (n.d.); Kbh3rd (2008); Rideout (2002).



Figure 4.9 (Continued)



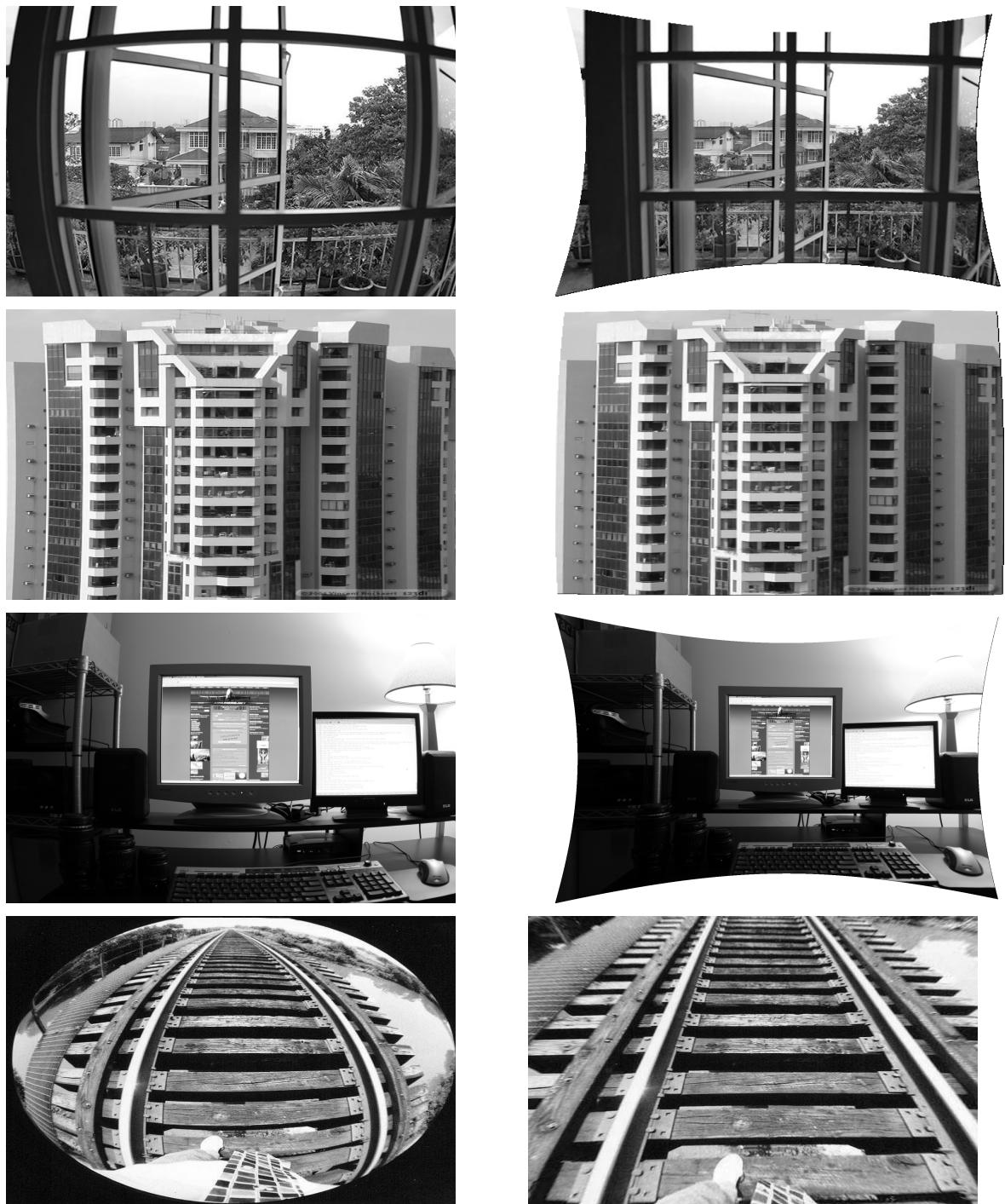


Figure 4.9 (Continued)



Figure 4.9 (Continued)



Figure 4.9 (Continued)

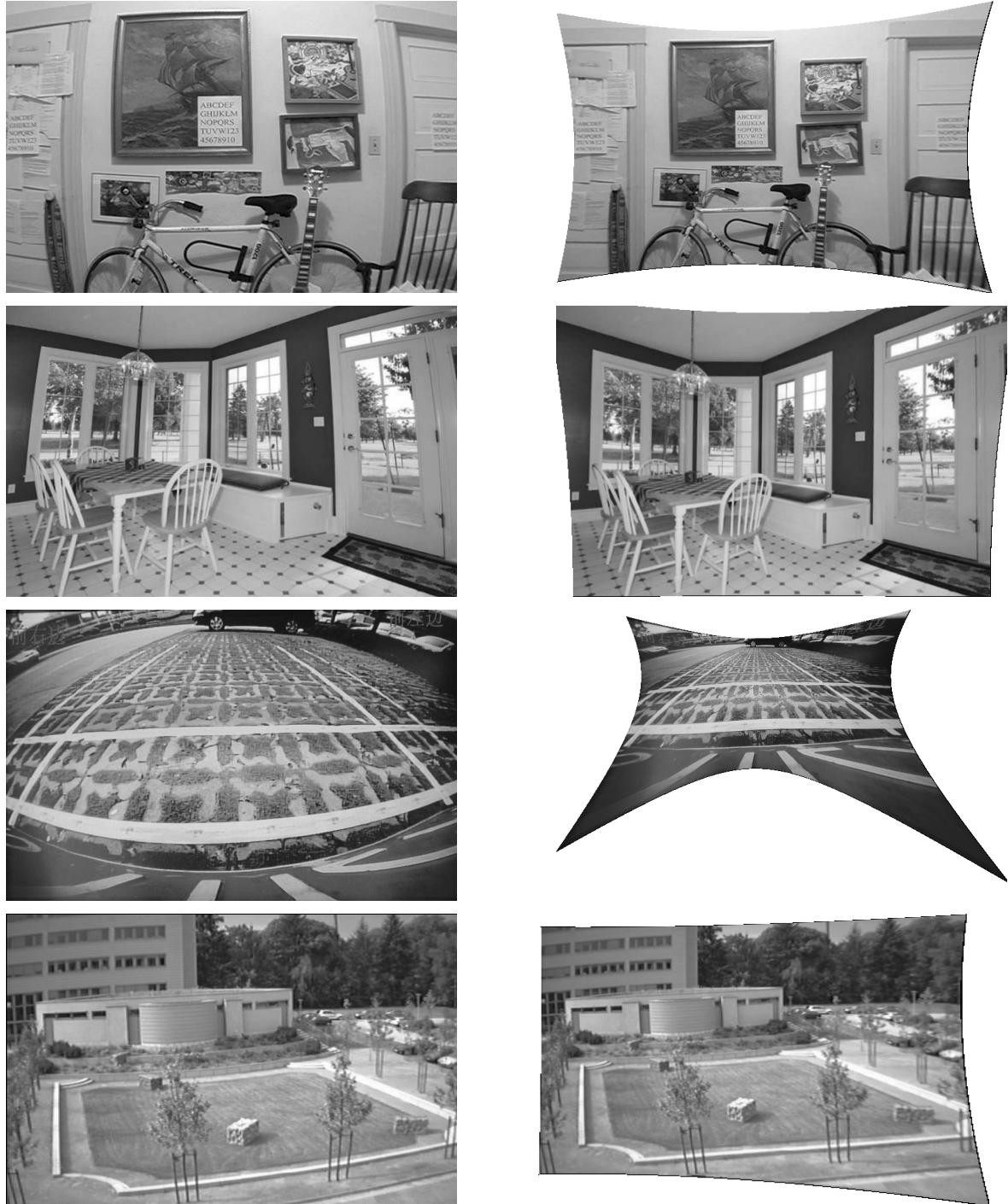


Figure 4.9 (Continued)



Figure 4.9 (Continued)

4.3 Failure case analysis

In some cases like in Figure 2.2 and Figure 4.9 (a) our algorithm fails to find the optimum results. I will briefly discuss this case. Actually both the images Figure 2.2 and Figure 4.9 (a) under perform due to a similar reason. In Figure 4.9 (a), I think the lines are too short to get an accurate estimate of the curvature. The problem is that the extracted lines (distorted arcs) in Figure 4.9 (a) are not only small in length but also contain less curvatures and based on these arcs, our algorithm is unable to find good estimates of the radial distortion parameters. As a consequence, the image reconstruction is poor. See the poor reconstruction of undistorted images in Figure 4.9 (b) and Figure 4.9 (c). The purpose of the MIN_CONTOUR_LENGTH at line number 326 in the DistortionEstimation.cpp in our source code is to filter small contours that usually are the source of the bad estimates.

I have run different experiments by changing the MIN_CONTOUR_LENGTH to get more arcs from the image. Consequently, I am getting more arcs but their curvatures and lengths are small that the algorithm is unable to find good estimates of the distortion parameters. Following are the results of the experiments by changing MIN_CONTOUR_LENGTH threshold:

- When the MIN_CONTOUR_LENGTH, sets at 10 pixels for the said image, then the detected arcs are 400.
- When the MIN_CONTOUR_LENGTH, sets at 20 pixels for the said image, then the detected arcs are 295.
- When the MIN_CONTOUR_LENGTH, sets at 30 pixels for the said image, then the detected arcs are 174.
- When the MIN_CONTOUR_LENGTH, sets at 40 pixels for the said image, then the detected arcs are 35.
- When the MIN_CONTOUR_LENGTH, sets at 50 pixels for the said image, then the detected arcs are 2.

One would have to link the sides of different squares into a single contour to get good estimates of the distortion in this case, but that is beyond the scope of this work.

4.4 Comparison with Alvarez et al. (Alvarez et al., 2009) method

How do the results presented thus far compare to previous work? Since there is no standardized database with ground truth for radial distortion correction, it is difficult to say. However, Álvarez et al. León et al. (2011) have deployed an excellent demo Web site for their method that allow users to submit an image for undistortion after manually selecting distorted lines from it. The method is also plumb-line based, with source code available online. We selected two synthetic images as shown in Figure 4.4 (a) image and Figure 4.5 (f) for comparison with their method. In the first image, the distortion center is $(320, 240)$ and $\lambda = -10^{-6}$; in the second image we moved the distortion center to $(390, 310)$ but kept λ fixed. For fair comparison, before submission, we selected the same number of lines while comparing our

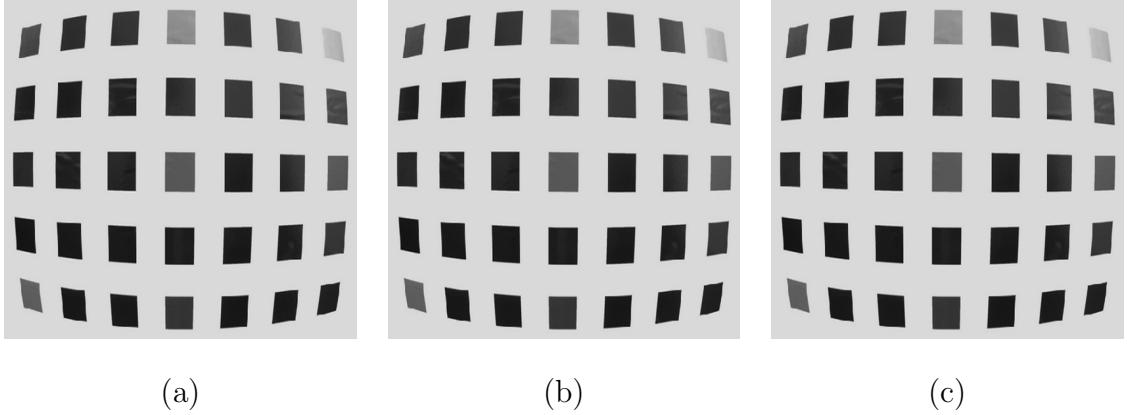


Figure 4.9: Failure case analysis: (a) Original image. (b) Undistorted image by setting the MIN_CONTOUR_LENGTH to 25 pixels. (c) Undistorted image by setting the MIN_CONTOUR_LENGTH to 30 pixels.

Table 4.1: Comparison with Alvarez et al. Alvarez et al. (2009). Image size is 640×480 . Distortion center = DC, RMSE (pixel intensity)= RMSE, Cpu time = Time, and Total number of arcs = Arcs.

Method	DC	RMSE	PNSR (dB)	Time (s)	Arcs
Álvarez et al. (2009)	320×240	4.6274	34.8241	1.219 ¹	81
	390×310	8.5204	29.5216	1.307 ¹	73
Our method	320×240	3.86525	36.3872	6.0463 ²	81
	390×310	3.91738	36.2709	5.2171 ²	73

1. Includes network latency but does not include time to manually select contours.

2. Includes time to perform edge processing, automatically select circular contours, and estimate arcs.

method with their method. To achieve this, we ran our algorithm to determined how many arcs are extracted by our method, then based on that we selected the same number of distorted lines, and as much as possible number of pixels for the Álvarez et al. method, and after obtaining the results, we manually scaled and translated the resulting images to be in the best possible alignment with the original undistorted image. Table 2 presents the results. When the image center is the distortion center, the Canny edge execution time is 0.6539s, arc detection takes 1.5945s, distortion parameter estimation time is 3.7980s, and the total execution time is 6.0463s. When the image center is not the distortion center, the Canny edge execution time is 0.6542s, arc detection takes 1.6309s, distortion parameter estimation time is 2.9320s, and the total execution time is 5.2171s. The Álvarez et al. method runs faster, even taking the Web service’s network latency into account, but requires manual intervention to select distorted straight lines. It took us around 1 hour to select 81 contours using their Web demo. When the image center is the distortion center, the Álvarez et al. method performs well, with a moderate 19.718% increase in RMSE and 1.5 dB decrease in

PNSR compared to our method. When the distortion center is not the image center, our method degrades only slightly, but the Álvarez et al. method degrades substantially. This is expected because the Álvarez et al. method is not designed to handle distortion centers that are not the image center. But our method still performs slightly better than the Álvarez et al. method even when the image center *is* the distortion center, even though our algorithm is estimating rather than assuming the location of the image center and even though the method is fully automatic with no user intervention.

Chapter 5

Conclusion

In this dissertation, we have introduced a new algorithm for automatic radial distortion estimation and removal using the plumb-line approach. The method works from a single image and does not require a special calibration pattern. It is based on Fitzgibbon's division model, robust estimation of circular arcs, and robust estimation of distortion parameters. As our method is based on circles, we also provide a detailed study of circle fitting methods and have found that two circle fitting methods, namely "Ransac-LM" and "Ransac-Pratt" perform better than the remaining 8 algorithms. "Ransac-Pratt" is a non-iterative circle fitting method and performs nearly as well as "Ransac-LM". Since "Ransac-Pratt" is computationally cheaper than "Ransac-LM," it may be recommended for applications in which runtime is important. Robust automatic radial distortion estimation from a single natural image would be extremely useful for many applications, particular those in human-made environments containing abundant lines. For example, it could be used to get a mobile robot or quadrotor experiment up and running quickly in an indoor environment. In a series of experiments on synthetic and challenging real images, we have demonstrated the method's ability to accurately identify distortion parameters and remove distortion from images. Data and source code based on OpenCV (Bradski, 2000) is available online¹ for researchers interested in evaluating or extending our procedure.

¹See <http://www.cs.ait.ac.th/vgl/faisal/downloads.html>.

References

- Al-Sharadqah, A., & Chernov., N. (2009). Error analysis for circle fitting algorithms. *The Electronic Journal of Statistics*, 3, 886-911.
- Alvarez, L., Gomez, L., & Sendra, J. R.(2009). An algebraic approach to lens distortion by line rectification. *Journal of Mathematical Imaging and Vision*, 39(1), 36–50.
- Alvarez, L., Gomez, L., & Sendra, J. R. (2010). *Algebraic lens distortion model estimation*. (http://www.ipol.im/pub/algo/ags_algebraic_lens_distortion_estimation/)
- Barreto, J. P., & Daniilidis, K. (2005). Fundamental matrix for cameras with radial distortion. In *International conference on computer vision (iccv)* (pp. 625–632).
- Bockaert, V. (n.d.). *Pincushion distortion*. (http://www.dpreview.com/learn/?/Glossary/Optical/Pincushion_Distortion_01.htm)
- Bouguet, J.-Y. (2008). *Camera calibration toolbox for matlab*. The MathWorks Inc. (http://www.vision.caltech.edu/bouguetj/calib_doc/index.html)
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Braüer-Burchardt, C.(2004). A simple new method for precise lens distortion correction of low cost camera systems. In *German pattern recognition symposium* (pp. 570–577).
- Brauer-Burchardt, C., & Voss, K. (2001). A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images. In *Ieee international conference on image processing* (Vol. 1, p. 225 - 228).
- Brown, D. C. (1971). Close-range camera calibration. *Photogrammetric Engineering*, 37(8), 855–866.
- Bucket, P. (n.d.). *Nikon 16mm fisheye photos*. (<http://photobucket.com/images/Nikon+16mm+fisheye/>)
- Bukhari, F., & Dailey, M. N.(2010). Robust radial distortion from a single image. In *Proceedings of the 6th international conference on advances in visual computing - volume part ii* (pp. 11–20).

- Bukhari, F., & Dailey, M. N. (2012). Automatic radial distortion estimation from a single image. *Journal of Mathematical Imaging and Vision*.
- Chanel. (n.d.). *How to fake the fisheye effect.* (<http://chanelwood.com/how-to/photoshop/how-to-fake-the-fisheye-effect/>)
- Chen, P.-Y., Huang, C.-C., Shiau, Y.-H., & Chen, Y.-T. (2009, January). A vlsi implementation of barrel distortion correction for wide-angle camera images. *Trans. Cir. Sys.*, 56, 51–55.
- Chernov, N. (1997). *Matlab code for circle fitting algorithms.* <http://www.math.uab.edu/~chernov/c1/MATLABcircle.html>.
- Chernov, N. (2010). *Circular and linear regression: Fitting circles and lines by least squares.* Chapman & Hall.
- Chernov, N., & Lesort, C. (2005, November). Least squares fitting of circles. *J. Math. Imaging Vis.*, 23, 239–252.
- Cyganek, B., & Siebert, J. (2009). *An introduction to 3D computer vision techniques and algorithms* (First ed.). Wiley and Sons.
- Devernay, F., & Faugeras, O. (2001). Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments. *Machine Vision and Applications*, 13(1), 14–24.
- Dyer, D. (n.d.). *Wide angle adapters for digital cameras.* (<http://www.andromeda.com/people/ddyer/photo/wideangle.html>)
- El-Melegy, M. T., & Farag, A. A. (2003). Nonmetric lens distortion calibration: Closed-form solutions, robust estimation and model selection. In *Computer vision, ieee international conference on* (Vol. 1).
- Faugeras, O. D., Luong, Q.-T., & Maybank, S. J. (1992). *Camera Self-Calibration: Theory and Experiments.* Springer-Verlag.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Fitzgibbon, A. W. (2001). Simultaneous linear estimation of multiple view geometry

- and lens distortion. In *Ieee international conference on computer vision and pattern recognition (cvpr)* (pp. 125–132).
- Friel, M., Hughes, C., Denny, P., Jones, E., & Glavin, M. (2010, June). Automatic calibration of fish-eye cameras from automotive video sequences. *Intelligent Transport Systems, IET*, 4(2), 136–148.
- Gonzalez-Aguilera, D., Gomez-Lahoz, J., & Rodriguez-Gonzalvez, P. (2011). An automatic approach for radial lens distortion correction from a single image. *IEEE Sensors Journal*, 11(4).
- Hartley, R. I., & Kang, S. B. (2005). *Parameter-free radial distortion correction with center of distortion estimation*.
- Hartley, R. I., & Kang, S. B. (2007, august). Parameter-free radial distortion correction with center of distortion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8), 1309–1321.
- Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2nd ed.). Cambridge University Press.
- Hughes, C., Glavin, M., Jones, E., & Denny, P. (2009, March). Wide-angle camera technology for automotive applications: a review. *Intelligent Transport Systems, IET*, 3(1), 19–31.
- Kang, S. B. (1997). *Semiautomatic methods for recovering radial distortion parameters from a single image*.
- Kang, S. B. (2000). Radial distortion snakes. *IEICE TRANSACTIONS on Information and Systems*, 1603–1611.
- Kbh3rd. (2008). camelback locomotive. (http://en.wikipedia.org/wiki/File:B%2B0_173_camelback_locomotive.jpg)
- Kukelova, Z., & Pajdla, T. (2011). A minimal solution to radial distortion autocalibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(1).
- Kukush, A., Markovsky, I., & Van Huffel, S. (2004). Consistent estimation in an implicit quadratic measurement error model. *Computational Statistics & Data Analysis*, 47(1), 123–147.

- Lakssi, L. (n.d.). *Motorcycle taxi*. (<http://leolaksi.wordpress.com/2009/12/28/more-photos-with-the-nikkor-16mm-f2-8-fisheye-lens/>)
- Lenz, R., & Tsai, R. (1988). 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, 713-720.
- León, L. Á., Déniz, L. G., & Castellano, P. H. (2011). Zoom Dependent Lens Distortion Mathematical Models. *Journal of Mathematical Imaging and Vision*.
- MATLAB. (2009). *edge - find edges in grayscale image*. The MathWorks Inc. (<http://www.mathworks.com/help/toolbox/images/ref/edge.html>)
- Moons, T., Gool, L. J. V., & Vergauwen, M. (2009). 3d reconstruction from multiple images: Part 1 - principles. *Foundations and Trends in Computer Graphics and Vision*, 4(4), 287-404.
- Moré, J. J. (1978). The Levenberg-Marquardt algorithm: Implementation and theory. *Lecture Notes in Mathematics*, 105–116.
- Ociepka, R. (2003). *Raynox dcr-720 with barrel distortion*. (<http://www.pbase.com/ociepka/image/32663644>)
- Oleson, R. (n.d.). *Full-circle examples*. (<http://rickoleson.tripod.com/index-105.html>)
- Pratt, V. (1987, August). Direct least-squares fitting of algebraic surfaces. *SIGGRAPH Comput. Graph.*, 21, 145–152.
- Ramalingam, S., Sturm, P., & Lodha, S. K. (2010). Generic self-calibration of central cameras. *Computer Vision and Image Understanding*, 114(2), 210–219.
- Rideout, S. (2002). *10 ninjas steve's photostream*. (<http://www.flickr.com/photos/steverideout/50185284/>)
- Rosten, E., & Loveland, R. (2011, January). Camera distortion self-calibration using the plumb-line constraint and minimal hough entropy. *Mach. Vision Appl.*, 22, 77–85.
- Sarge. (n.d.). *Misc*. (http://sarge.wheresthebeef.co.uk/Misc/350D_misc_100/IMG_2797.jpg)

- Skewes, K. (n.d.). *Real estate photo tip: Wide angle lens correction.* (<http://blogonlineed.com/2011/01/14/real-estate-photo-tip-wide-angle-lens-correction/>)
- Solheim, E. (n.d.). *How to remove distortion on a fisheye image.* (<http://eirikso.com/2008/12/14/how-to-remove-distortion-on-a-fisheye-image/>)
- Stein, G. P. (1996). Lens distortion calibration using point correspondences. In *Ieee international conference on computer vision and pattern recognition (cvpr)* (pp. 602–608).
- Strand, R., & Hayman, E. (2005). Correcting radial distortion by circle fitting. In *British machine vision conference (bmvc)*.
- Swaminathan, R., & Nayar, S. (2000, Oct). Non-Metric Calibration of Wide-Angle Lenses and Polycameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10), 1172 – 1178.
- Taubin, G. (1991). Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11), 1115–1138.
- Tavakoli, H. R., & Pourreza, H. R. (2010). Automated center of radial distortion estimation, using active targets. In *Asian conference on computer vision (accv)*.
- Thormählen, T., Broszio, H., & Wassermann, I. (2003). Robust line-based calibration of lens distortion from a single view. In *Computer vision / computer graphics collaboration for model-based imaging rendering, image analysis and graphical special effects* (pp. 105–112).
- Tomasi, C. (2007). *Sample image for CPS 296.1 homework assignment.* (<http://www.cs.duke.edu/courses/spring06/cps296.1/homework/1/lab.gif>)
- Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Radiometry*, 221–244.
- Wang, A., Qiu, T., & Shao, L. (2009). A simple method of radial distortion correc-

- tion with centre of distortion estimation. *Journal of Mathematical Imaging and Vision*, 35(3), 165–172.
- Wang, J., Shi, F., Zhang, J., & Liu, Y. (2008, February). A new calibration model of camera lens distortion. *Pattern Recogn.*, 41, 607–615.
- Whittaker, G. (n.d.). *Wide angles*. (<https://picasaweb.google.com/gmw3027/WideAngles#5276761797451570738>)
- Yusuf. (n.d.). *Correcting barrel distortion of wide and ultrawide lenses*. (<http://www.photos-of-the-year.com/articles/barrel-distortion/>)
- Zhang, Z. (2000, November). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334.