CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY

RESEARCH REPORT

Correcting Radial Lens Distortion without Knowledge of 3-D Structure

Tomáš Pajdla
Tomáš Werner
Václav Hlaváč

N° K335-CMP-1997-138

December 18, 1997

Czech Technical University, Faculty of Electrical Engineering
Department of Control Engineering, Center for Machine Perception
121 35 Prague 2, Karlovo náměstí 13, Czech Republic
fax +420 2 24357385, phone +420 2 24357465, http://cmp.felk.cvut.cz

Acknowledgement

This research was supported by the Czech Ministry of Education grant VS96049, the Grant Agency of the Czech Republic, grants 102/95/1378, 102/97/0480, 102/97/0855 and European Union grant Copernicus CP941068.

Contents

1 Introduction **1**

2 Camera and Distortion Model **2**
2.1 Linear camera 2
2.2 Radial lens distortion 2
2.3 Inverse to radial lens distortion 3

3 Estimation of Radial Distortion **5**
3.1 Estimation from coplanar points known up to a homography 5
3.2 Estimation from straight lines 6
3.3 Scenaria for estimation from straight lines 8

4 Experiments **9**
4.1 Experiments on simulated data 9
4.2 Experiments on real data 9
4.2.1 Scenes 9
4.2.2 Line extraction 10
4.2.3 Results 10

5 Validity of radial distortion model for the scenes with large depth variations **16**

6 Conclusions **19**
6.1 Other relevant literature 19

A Programmer's Guide to the RADIAL Toolbox **20**

Chapter 1

Introduction

Most works on geometrical 3-D vision assume a *pinhole camera*. This camera projects the world to the image plane *linearly* if expressed in terms of projective geometry. Thanks to this linearity, a lot of considerations on geometry in which cameras are involved become simple.

However, real cameras are not linear. The main reason is that the optical system of the camera is not linear. The camera model is then possible to assume as a composition of a linear camera and a non-linear distortion(s). In many cases, the influence of the non-linear distortion can be neglected and the camera considered linear. In cases where the imaging geometry has to be known with high accuracy, the non-linear distortion must be modeled and estimated.

Usually, the non-linear distortion is estimated along with all (extrinsic and intrinsic) parameters of the linear camera. This is done using a set of correspondences [2-D image point, 3-D scene point]. The disadvantage is that obtaining accurate coordinates of 3-D scene points is sometimes demanding or impossible. A sufficiently robust algorithm for estimating the same complete set of calibration parameters without knowledge of 3-D scene points is not known.

Non-linear lens distortions are of several kinds. The most pronounced one is *radial distortion*. In our work, we propose to estimate and compensate for the radial lens distortion *without* full knowledge of 3-D point coordinates. In this case, only radial distortion parameters are estimated, unlike the extrinsic and intrinsic parameters of the linear camera. The radial distortion parameters are estimated accurately enough to "remove" the radial distortion and thus obtain a linear camera. Yet, their accuracy need not be sufficient for other purposes.

We propose two different ways for correcting the radial distortion. The first makes use of knowledge that certain segments in images are straight lines in the scene. The second assumes a coplanar set of points in the scene known up to an unknown plane-to-plane homography.

In this text, the proposed algorithms and necessary background are described in detail. Moreover, the guide to the Radial Distortion Toolbox for MATLAB is included. This toolbox is the implementation of the developed distortion models and correction algorithms.

Chapter 2

Camera and Distortion Model

2.1 Linear camera

We consider the model of a linear, pinhole camera in the form[1]

$$\begin{bmatrix} \mathbf{u}_l \\ 1 \end{bmatrix} \simeq \mathbf{K}[\mathbf{R} \mid -\mathbf{R}\mathbf{t}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} , \tag{2.1}$$

where $\mathbf{u}_l = [u_l \ v_l]^\top$ are the coordinates of an image point, being projection of a scene point with coordinates $\mathbf{X} = [X \ Y \ Z]^\top$. $\mathbf{K}$ is the calibration matrix, containing all intrinsic camera parameters. $\mathbf{R}$ and $\mathbf{t}$ are rotation matrix and translation vector, the extrinsic camera parameters.

This camera is an abstraction, $\mathbf{u}_l$ is not directly measurable.

2.2 Radial lens distortion

Radial lens distortion originates from the fact that no lens, no matter if with spherical or any other surface, images linearly. Unlike some other types of lens distortion, the radial distortion is thus not due to imprecise manufacturing but rather an inherent property of any lens.

Let us consider the radial distortion generally as

$$\mathbf{u} = d(\mathbf{u}_l, \mathbf{p}) , \tag{2.2}$$

where $\mathbf{u} = [u \ v]^\top$ are the coordinates of a distorted image point, for $\mathbf{u}_l$ see Eq. 2.1, d is the distortion function, and $\mathbf{p}$ is the vector of distortion parameters. $\mathbf{u}$ is directly measurable eg as distances at a physical photographic film or coordinates of a pixel in an image from a CCD camera.

The model of radial distortion is expressed by the following equations:

$$\frac{r}{r_l} = \frac{||\mathbf{u} - \mathbf{u}_0||}{||\mathbf{u}_l - \mathbf{u}_0||} = \frac{u - u_0}{u_l - u_0} = \frac{v - v_0}{v_l - v_0} = D(r_l, \mathbf{k}) , \tag{2.3}$$

where the polynom on the right-hand side is given by

$$D(r_l, \mathbf{k}) = 1 + k_1 r_l^2 + k_2 r_l^4 + \ldots + k_n r_l^{2n} . \tag{2.4}$$

The point $\mathbf{u}_0 = [u_0 \ v_0]^\top$ is the image center (principal point), $\mathbf{k} = [k_1 \ k_2 \ \ldots \ k_n]^\top$ are the distortion coefficients, and $r_l = ||\mathbf{u}_l - \mathbf{u}_0|| = [(u_l - u_0)^2 + (v_l - v_0)^2]^{1/2}$ is the radius of $\mathbf{u}_l$ or distance of $\mathbf{u}_l$ from $\mathbf{u}_0$. The length of the vector $\mathbf{k}$ is denoted by n, and $2n$ is the order of the distortion polynomial. For most of practical tasks, second order ($n = 1$) or fourth order ($n = 2$) are sufficient.

[1] We use the notations $[\mathbf{u} \ 1]^\top$ and $[\mathbf{X} \ 1]^\top$ for a projective point for the better clarity of the text. However, the notations $[u \ v \ w]^\top$ and $[X \ Y \ Z \ W]^\top$ would be more precise because they allow ideal points.

These equations imply the following expression for the function d:

$$\mathbf{u} = d(\mathbf{u}_l, \mathbf{p}) = \mathbf{u}_0 + (\mathbf{u}_l - \mathbf{u}_0)D(r_l, \mathbf{k}) \ , \tag{2.5}$$

Note that $\mathbf{p} = [\mathbf{u}_0^\top \ \mathbf{k}^\top]^\top$.

Could the model of the radial distortion be different from that expressed by Eqs. 2.3 and 2.4? The general form of the function can be obtained if we formulate reasonable requirements it should satisfy:

1. The function is radially symmetric around the image center, $\mathbf{u}_0$. Hence it is expressible in terms of radius r_l only.
2. The function is odd.
3. The function is continuous, hence it is zero if $r_l = 0$.
4. The function is a polynomial. This requirement is not so obvious as the above remaining ones; it reflects the fact that an unknown underlying distortion function is approximated by Taylor polynomial.

Taking into account these requirements[2], we obtain for the most general case $r = r_l D(r_l, \mathbf{k})$ where $D(r_l, \mathbf{k})$ is an even function. Thus, the form of Eq. 2.3 is the most general one.

The polynom $D(r_l, \mathbf{k})$ embodies all the information about the radial distortion. It is an approximation of some underlying function by Taylor series. As the only requirement on it is that it be an even function, its most general form is slightly different from Eq. 2.4:

$$D(r_l, \mathbf{k}) = \alpha(1 + k_1 r_l^2 + k_2 r_l^4 + \ldots + k_n r_l^{2n}) \ . \tag{2.6}$$

Indeed, the photogrammetric community sometimes use

$$D(r_l, \mathbf{k}') = 1 + k_1'(r_l^2 - r_0^2) + k_2'(r_l^4 - r_0^4) + \ldots + k_n'(r_l^{2n} - r_0^{2n}) \ . \tag{2.7}$$

The parameters from Eq. 2.6 are related to the parameters from Eq. 2.7 by

$$\alpha = 1 - k_1' r_0^2 - k_2' r_0^4 - \ldots - k_n' r_0^{2n} \ , \tag{2.8}$$

$$\mathbf{k} = \frac{\mathbf{k}'}{1 - k_1' r_0^2 - k_2' r_0^4 - \ldots - k_n' r_0^{2n}} \ . \tag{2.9}$$

Eq. 2.6 is equivalent to Eq. 2.7 as long as Eq. 2.8 is soluable for r_0. Otherwise, Eq. 2.6 is more general.

Why is it sufficient to consider $D(r_l, \mathbf{k})$ as Eq. 2.4, rather than Eq. 2.6? The reason is that the parameter α adds only the possibility of *linear scaling* that is already comprised in the calibration matrix $\mathbf{K}$ in the linear camera model, Eq. 2.1. The simultaneous estimation of α and $\mathbf{K}$ would be ambiguous. It is natural to choose $D(r_l, \mathbf{k})$ such that $D(r_l, \mathbf{k}) = 1$ for $\mathbf{k} = \mathbf{0}$ (zero distortion), hence $\alpha = 1$ and Eq. 2.4.

2.3 Inverse to radial lens distortion

It is often required to compute $\mathbf{u}_l$ given $\mathbf{u}$ and $\mathbf{p}$. Then the inverse function to d (see Eq. 2.2) must be used. Let us denote it d_{-1},

$$\mathbf{u}_l = d_{-1}(\mathbf{u}, \mathbf{p}) \ . \tag{2.10}$$

For a general n, there does not exist a closed-form expression for d_{-1}. Therefore, we propose an iterative algorithm for evaluating d_{-1}, based on Banach's fixed point theorem [2].

It suffices to find r_l given r and $\mathbf{k}$ from Eq. 2.3 or, equivalently, to find r_l for which the function $f(r_l) = r - r_l D(r_l, \mathbf{k})$ vanishes. The algorithm works as follows:

[2]To be precise, the set of requirements is redundant. Instead of the requirements 2, 3, and 4 it suffices to say that the function be an odd polynomial.

1. Let $r_l := r$.
2. If $|f(r_l)| < \mathsf{tiny}$, stop (success, r_l is the result).
3. If $|f(r_l)| > \mathsf{huge}$, stop (failure due to divergence).
4. Let $r_l := r_l - \lambda f(r_l)$.
5. Goto 2.

If $\lambda = -1$, simple iteration algorithm is obtained. If $\lambda = 1/f'(r_l)$, Newton method is obtained.

An important issue is to find, under what conditions the above algorithm converges. Necessary and sufficient condition is that

$$\left|\frac{\mathrm{d}}{\mathrm{d}r_l}\left[r_l - \lambda f(r_l)\right]\right| < a \leq 1 \,, \tag{2.11}$$

it means that

$$\left|1 + \lambda\left[1 + 3k_1 r_l^2 + 5k_2 r_l^4 + \ldots + (2n+1)k_n r_l^{2n}\right]\right| < a \leq 1 \,. \tag{2.12}$$

For $\lambda = -1$ the condition 2.11 is satisfied if elements of $\mathbf{k}$ are "small". In practice, lens distortions are small and hence also elements of $\mathbf{k}$. Unfortunately, we are not able to formulate the convergence condition more exactly. We observed that the algorithm converges reliably for parameters of a real lens.

We observed that simple iterations (ie, $\lambda = -1$) needs about two to four times more iterations to converge than the Newton method (ie, $\lambda = 1/f'(r_l)$). As the Newton method needs additional evaluation of $f'(r_l)$, it brings no substantial advantage. The Newton method is also known to converge less reliably than simple iterations[3]. For these reasons, we advocate to use $\lambda = -1$. Typically, the algorithm then requires about 4 to 10 iterations to converge with accuracy almost equal to machine precision.

[3]This was claimed by the mathematician Mirko Navara from our research group. Currently, we cannot provide the reader with a more convincing proof of this assertion.

Chapter 3

Estimation of Radial Distortion

Let us assume we have a reasonable estimate of the radial distortion parameters, $\mathbf{u}_0$ and $\mathbf{k}$. Then it is possible to "undo" the distortion by transforming the images, captured by the non-linear camera, by d_{-1}. The transformed images will be as if captured by a linear camera.

Having estimated the radial distortion parameters $\mathbf{p}$, we have two possibilities how to use it to remove the distortion from captured images:

1. Detect features in the image. Transform positions of the features using Eq. 2.10.
2. Transform the image bitmap using Eq. 2.2 so that the transformed bitmap is as if captured by a linear camera[1]. Detect features in the transformed image.

We propose two different algorithms for estimating $\mathbf{u}_0$ and $\mathbf{k}$. They are described in the following two sections.

3.1 Estimation from coplanar points known up to a homography

Let us assume that there is a coplanar set of points in the scene, and that the coordinates of these points in the plane are known up to an unknown homographic transformation. This plane-to-plane homography relates the scene points, denoted by $\mathbf{u}_w$, with their images $\mathbf{u}_l$ in the image plane of the *linear* camera[2]:

$$\begin{bmatrix} \mathbf{u}_l \\ 1 \end{bmatrix} \simeq \mathbf{H} \begin{bmatrix} \mathbf{u}_w \\ 1 \end{bmatrix} . \tag{3.1}$$

The points $\mathbf{u}_l$ are not directly measurable. Only the distorted points, $\mathbf{u}$, obtained from Eq. 2.2, are measurable.

If the points $\mathbf{u}_w$ and $\mathbf{u}$ are known, the homography $\mathbf{H}$ and distortion parameters $\mathbf{p} = [\mathbf{u}_0^\top \ \mathbf{k}^\top]^\top$ can be computed from Eqs. 3.1 and 2.2.

The system of equations 3.1 and 2.2 is non-linear, such that there is no closed-form solution[3] for $\mathbf{H}$ and $\mathbf{p}$. Moreover, an overdetermined system has to be solved in a real situation because of noise in $\mathbf{u}_w$ and $\mathbf{u}$. Non-linear least squares, eg the Levenberg-Marquardt algorithm, is therefore a suitable approach.

Assume that a set of points $\mathbf{u}_w$ and their images $\mathbf{u}$ have been measured. The algorithm for computing the estimates $\mathbf{H}^*$ and $\mathbf{p}^*$ of $\mathbf{H}$ and $\mathbf{p}$ is the following:

1. Choose an initial value $\mathbf{p}_0$ of $\mathbf{p}$, near enough to the solution $\mathbf{p}^*$.

[1] Eq. 2.2 indeed has to be used rather than Eq. 2.10 because the scanning takes place in the *resulting* image.

[2] We can use the notation $[\mathbf{u}\ 1]^\top$ for a projective point since all considered points are finite.

[3] We have not proved that there is no closed-form solution, but its existence seems very unlikely.

2. Choose an initial value $\mathbf{H}_0$ of $\mathbf{H}$, near enough to the solution $\mathbf{H}^*$. If the radial distortion is "small", which is always the case in a real situation, $\mathbf{H}_0$ can be computed as the homography transforming $\mathbf{u}_w$ nearest to $\mathbf{u}$, eg in the least-squares sense. Expressed by a formula,

$$\mathbf{H}_0 = \arg\min_{\mathbf{H}} \sum_{i=1}^{N} ||\mathbf{u}_i - \mathbf{u}_{\mathbf{H}i}||^2 , \tag{3.2}$$

where

$$\begin{bmatrix} \mathbf{u}_{\mathbf{H}i} \\ 1 \end{bmatrix} \simeq \mathbf{H} \begin{bmatrix} \mathbf{u}_{wi} \\ 1 \end{bmatrix} . \tag{3.3}$$

The symbols with the subscript i refer to i-th point[4], the total number of points is N. The optimization problem 3.2 is linear and its solution is obvious.

3. Compute $\mathbf{H}^*$ and $\mathbf{p}^*$ as

$$\{\mathbf{H}^*,\ \mathbf{p}^*\} = \arg\min_{\mathbf{H},\,\mathbf{p}} \rho_{\text{homo}} , \tag{3.4}$$

where the objective function is given by

$$\rho_{\text{homo}} = \sum_{i=1}^{N} ||\mathbf{u}_i - \mathbf{u}'_i||^2 , \tag{3.5}$$

where $\mathbf{u}_i$ are obtained from $\mathbf{u}_{wi}$ using Eqs. 3.1 and 2.2, and $\mathbf{u}'_i$ are actually measured image points. The optimization problem 3.4 is solved by the Levenberg-Marquardt non-linear least squares algorithm, starting from the initial values $\mathbf{p}_0$ and $\mathbf{H}_0$.

The involved point coordinates should be approximately equal to one. It is required both for solving the problem Eq. 3.2 (for the least squares estimation to work properly) and for the problem Eq. 3.4 (for proper computing of finite differences in the Levenberg-Marquardt algorithm). It is advantageous to transform the measured points $\mathbf{u}_{wi}$ and $\mathbf{u}'_i$ before starting the algorithm to achieve better numerical stability, and to recompute $\mathbf{H}^*$ and $\mathbf{p}^*$ accordingly after exiting the algorithm. The points can be transformed eg in the sense of Hartley's normalization [3].

Is the solution unique[5], in other words, cannot the distortion described by $\mathbf{p}$ be somehow compensated by an appropriate change of $\mathbf{H}$? We are currently not able to give a rigorous proof but we think that the the solution is unique.

3.2 Estimation from straight lines

Let us assume that there are sets of points in the image and each set is known to lie at a straight line in the scene. The camera observes L straight lines in the scene, $\Lambda_1, \ldots, \Lambda_L$. At the k-th line, Λ_k, N_k scene points $\mathbf{X}^k_1, \ldots, \mathbf{X}^k_{N_k}$ are observed and projected to image points $\mathbf{u}^k_1, \ldots, \mathbf{u}^k_{N_k}$. If the radial distortion of the camera is non-zero (ie, $\mathbf{k} \neq \mathbf{0}$ in Eq. 2.5), the image points $\mathbf{u}^k_1, \ldots, \mathbf{u}^k_{N_k}$ generally do not lie on a straight line. However, the image points $\mathbf{u}^k_{l1}, \ldots, \mathbf{u}^k_{lN_k}$, projected by the underlying linear camera (see Section 2.1), *do* lie at a straight line. Again, only $\mathbf{u}^k_1, \ldots, \mathbf{u}^k_{N_k}$ are measurable.

We can estimate the distortion parameters $\mathbf{p}$ by requiring that $\mathbf{u}^k_{l1}, \ldots, \mathbf{u}^k_{lN_k}$ computed as

$$\mathbf{u}^k_{li} = d_{-1}(\mathbf{u}^k_i, \mathbf{p}) , \tag{3.6}$$

[4]The reader will forgive slightly unclear notation in the subscript of eg $\mathbf{u}_{wi}$, it means $(\mathbf{u}_w)_i$. We will use a similar notation in subscripts also further in the text.

[5]We call unique also the situation when $\mathbf{k} = \mathbf{0}$ even if $\mathbf{u}_0$ is not determined uniquely in this case.

where $i = 1, \ldots, N_k$, (see also Eq. 2.10) lie at straight lines for all $k = 1, \ldots, L$.

In a real situation, noise in the measured points requires to formulate the task as an optimization problem, similarly as in Section 3.1: Find such $\mathbf{p}$ that maximizes the straightness of the sets of points $\mathbf{u}^k_{l1}, \ldots, \mathbf{u}^k_{lN_k}$ for all lines Λ_k. For that, the notion of straightness has to be formalized.

We formalize straightness of a set of points $\mathbf{u}_1, \ldots, \mathbf{u}_N$ as a residual of fitting a straight line, $\mathbf{l} = [l_1\ l_2\ l_3]$, to the set (in fact, this defines *negative* straightness rather). The fitted line $\mathbf{l}^*$ minimizes the sum of squared distances from the points:

$$\mathbf{l}^* = \arg\min_{\mathbf{l}} \sum_{i=1}^{N} |\mathbf{l}, \mathbf{u}_i|^2 \ , \tag{3.7}$$

where $|\mathbf{l}, \mathbf{u}| = \mathbf{l}\mathbf{u}(l_1^2 + l_2^2)^{-1/2}$ is the Euclidean distance of the point $\mathbf{u}$ from the line $\mathbf{l}$. The straightness is defined as

$$\rho(\mathbf{u}_1, \ldots, \mathbf{u}_N) = -\sum_{i=1}^{N} |\mathbf{l}^*, \mathbf{u}_i|^2 \ . \tag{3.8}$$

Total straightness for all lines in the image, $\Lambda_1, \ldots, \Lambda_L$, could be defined as $\sum_{k=1}^{L} \rho(\mathbf{u}^k_{l1}, \ldots, \mathbf{u}^k_{lN_k})$. However, the problem with this definition is that it is not scale-invariant. Straightness cannot be made scale-invariant for each line separately (ie, separately for each ρ_k) but rather for all lines together. We therefore suggest to transform all the points $\mathbf{u}^k_{li}$ by a homography such that they are near to the measured points $\mathbf{u}^k_i$, and then to compute the total straightness, ρ_{lines}, from these transformed points:

$$\rho_{\text{lines}} = \sum_{k=1}^{L} \rho(\widehat{\mathbf{u}}^k_{l1}, \ldots, \widehat{\mathbf{u}}^k_{lN_k}) \ , \tag{3.9}$$

$$\begin{bmatrix} \widehat{\mathbf{u}}^k_{li} \\ 1 \end{bmatrix} \simeq \widehat{\mathbf{H}} \begin{bmatrix} \mathbf{u}^k_{li} \\ 1 \end{bmatrix} \ . \tag{3.10}$$

$$\widehat{\mathbf{H}} = \arg\min_{\mathbf{H}} \sum_{k=1}^{L} \sum_{i=1}^{N_k} ||\mathbf{u}^k_i - \widehat{\mathbf{u}}^k_{li}||^2 \ , \tag{3.11}$$

There are a lot of other ways how to define $\rho(\mathbf{u}_1, \ldots, \mathbf{u}_N)$ and ρ_{lines}, as well as how to enforce scale invariance. The suggested way has been observed to behave satisfactorily in experiments.

Assume that the points $\mathbf{u}^k_i$ have been measured. The algorithm for computing the estimate $\mathbf{p}^*$ of $\mathbf{p}$ looks like follows:

1. Choose the initial value $\mathbf{p}_0$ of $\mathbf{p}$, near enough to the solution $\mathbf{p}^*$.
2. Compute $\mathbf{p}^*$ as

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \rho_{\text{lines}} \ , \tag{3.12}$$

   where the objective function ρ_{lines} is given by Eq. 3.9. The optimization problem 3.12 is solved by the Levenberg-Marquardt non-linear least squares algorithm, starting from the initial value $\mathbf{p}_0$.

Like in the algorithm in Section 3.1, the involved point coordinates should be normalized to values near one.

To investigate the uniqueness of the solution is more difficult than in Section 3.1, at least because it depends on the number and location of the straight lines. Again, we are currently not able to give any proof but we will describe only what we have observed in the experiments. If the image was "densely enough" covered by lines and those were sampled by "enough" points, the solution seemed to be unique. If all the lines were parallel, the solution seemed to be ambiguous. If the points at the lines formed a grid, the solution seemed to be ambiguous if the grid was smaller than about 4×4 points.

3.3 Scenaria for estimation from straight lines

There exist several different scenarios allowing for radial distortion estimation from straight lines.

The simplest but most artificial is the scenario when the camera looks at an artificial calibration pattern like a checkerboard.

Different scenaria follows from the fact that the distortion is corrected only on the basis of line straightness. Therefore, it is possible to combine data from more images taken with the same camera. For instance, two images, one with the horizontal and one with the vertical lines can be shown to the camera.

Similarly, if one camera is moving and taking a sequence of images along its path, the lines extracted from all images in the sequence can be used for identification of the cameras' radial distortion. This procedure alleviates the difficulty with having many good lines in each calibration image. It is important for practical use as it is much easier to see a good line from time to time than to have a sufficient number of good lines in even one view. This opens possibility to correct camera distortion in older sequences where usually no calibration has been done at the time of their acquisition. It is enough to extract a few good lines.

A bit different scenario of calibration can also be proposed when the camera can be controlled at the time of sequence acquisition. Then, just one line can be taken several times so that it appears in the images in various orientations and on various positions. Such a line plays the role of a natural calibration pattern and replaces artificial patterns like a checkerboard.

Two same cameras can be corrected at once just by using the lines from both of them. By doing so, the same distortion model is imposed on both cameras. Such scenario is important if removing distortion from a stereo rig.

Chapter 4

Experiments

Experiments on simulated and real data have been performed. Both algorithms (Sections 3.1 and 3.2) have been tested extensively, the stress being put on algorithm 3.2 (estimation from lines).

4.1 Experiments on simulated data

The captured image was simulated as a set of points forming a regular grid, undergone further a slight homography (ie, a slight tilting of the grid). The number of the points was 16 to 100 (grid size 4×4 to 10×10).

Several distortion polynomials were simulated, from those causing a slight distortion, as is the case in a real situation, to those causing much larger distortions than it could arise for a real lens.

For the gradient descent algorithms, the initial values of the coefficients of the distortion polynomial were $\mathbf{k} = \mathbf{0}$, the initial value of the principal point, $\mathbf{u}_0$, were chosen in the geometrical center of the image and varied by $-50 + 33\%$.

For noiseless data, the algorithms quickly converged always to the correct estimation. The estimates were correct with error smaller than 10^{-8} after approx. 10 to 30 iterations.

An additive noise with uniform and Gaussian distribution was added to the image point coordinates of the captured images. The influence of the noise on the linearity of the "undistorted" lines/points was observed approximately *linear*, for the portions of the added noise as large as errors caused by the distortion (it makes no sense to add larger portions of noise).

For these noisy data, we guess that in one case out of several tens, the gradient descent converged to a false minimum, mainly if the initial estimate of the principal point was far from the actual one. This behaviour could be recognized according to significantly slower convergence speed than when the correct minimum was being reached.

The detailed description of the experiments can be found in the help text to the function `simul`, see Appendix.

4.2 Experiments on real data

4.2.1 Scenes

Three different scenes were captured by the same camera with the same 12 mm lens. The first scene, a checkerboard placed to be parallel to the image plane, is labeled by "Pattern A" and has been captured four times from the same viewpoint, Fig. 4.1(a). All images were taken with the same parameters of the camera just a few seconds one after the other.

The second scene, slanted checkerboard, is labeled by "Pattern B" and has been captured twice from the same viewpoint, Fig. 4.2(a). The parameters of the camera, namely the focus, has been changed since taking the images of the "Pattern A".

The third scene, the door of our laboratory, labeled by "doors" has been captured four times while changing the viewpoint by moving and rotating the camera to collect enough linear segments, Fig. 4.3. The label "doors: view i + j + ..." means that the model has been estimated from all the data extracted from all the images i, j, ... The parameters of the camera has probably also been changed since taking the images of the "Pattern B" in this case.

4.2.2 Line extraction

In the case of the checkerboard scenes, ie Pattern A and Pattern B, the lines, Fig. 4.1(b) and Fig. 4.1(b), were extracted automatically in the following way:

1. Deriche edge detector has been applied to rows and columns of the images independently resulting in the set of edge pixels.

2. Hough transform has been used to extract the line parameters independently for the vertical and the horizontal lines. Each line has been assigned a number to.

3. The edge pixels were labeled by the number equal to the number of the closest line. Pixels with the same number are treated as the points of a line, respectively of a distorted line.

   In fact, there were two families of line segments resulting from the rising and falling edges. These edges has been detected with slight relative shift however they should lie on the same line. In other words, our edge detector found black squares to be smaller than the white ones.

   At the moment, we are not able to explain this phenomenon and therefore we were not able to compensate for it. We have a theory, that the black/white transition has too high contrast so that our CCD camera was not able to "squeeze" the signal into its dynamic range. It would cause signal to be truncated on the low levels as we closed the iris in order not to get into the saturation on high levels. The truncation of low levels would bring assymetry biasing edge localization.

   On the other hand, the bias seems to be a kind of a translation, ie linear operation, that should not affect linearity of the camera. Therefore, we have split the detected segments into two families. One family consists of segments formed by edge pixels detected on rising edges while the other one is formed by those detected on the falling edges. Though these lines are altogether affected by some nonlinear action, when treated separately each is, as we hope, affected by a translation.

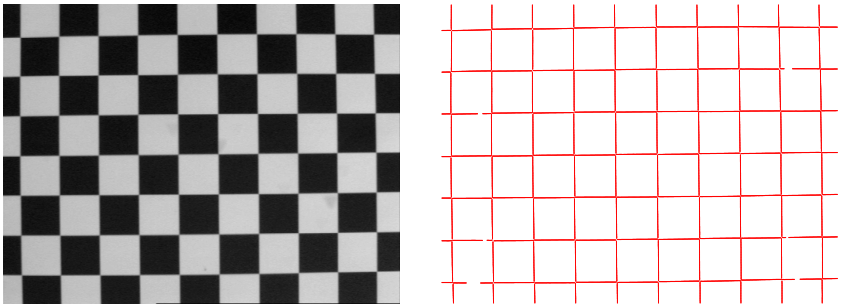
Lines in real scene of the door, Fig. 4.3, were extracted by picking all the edge points surrounded by manually defined rectangles.

4.2.3 Results

When evaluating the results visually, all the lines were corrected well. Figure 4.6 and Figure 4.7 show distorted and corrected calibration patterns. Figure 4.4 and Figure 4.5 show a detail of a few corrected lines of the calibration pattern A. Similar results for the real scene are shown on Figure 4.8. All the above results are shown for the correction by the second order distortion model (ie, $n = 1$ in Eq. 2.4).

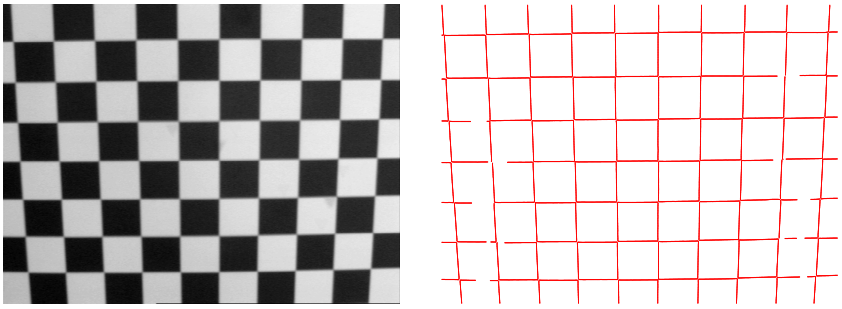
Figure 4.9 compares the results numerically and graphically for various scenes, images, and distortion models.

Distortion polynomials 2.4 up to the second ($n = 1$) degree (left column of the figures) and up to the fourth ($n = 2$) degree (right column of the figures) degree were estimated and compared.



(a) (b)

Figure 4.1: (a) Gray level image of the calibration pattern A and (b) the extracted edges. The checkerboard is perpendicular to the optical axis of the camera.



(a) (b)

Figure 4.2: (a) Gray level image of the calibration pattern B and (b) the extracted edges. The checkerboard is slanted.

Tables 4.9(a) and 4.9(b) show the parameters of the models estimated in all the situations. In vector $[e, eu, e/eu, u0, v0, k1, k2]$, e in [mm] stands for the overall error of line fit before the distortion correction, eu in [mm] stands for the overall error of line fit after the correction, the ratio e/eu expresses the improvement of the linearity of the camera after the correction, and $u0, v0, k1, k2$ are the parameters of the distortion polynomial.

Figures 4.9(c) and 4.9(d) show the plots of the distortion as a function of the radius. The view 1, 3, and 4 delivered almost identical results independently from the order of the model.

The view 2 provided slightly different estimates. After closer investigation of the extracted data, we have found that a small false edge has been extracted from the images. However, it has not been proved that that this error really caused the variation.

Figures 4.9(e) and 4.9(f) show similar results for the Pattern B.

Figures 4.9(g) and 4.9(h) show the error functions for the reals scene. It is apparent that there is a large difference in results obtained from the view 1 and the view 2. Also, while for the calibration patterns, there were almost no differences between the estimates for different orders, here the second order differs quite a lot. The largest difference is in the view 1 what suggests that the results could be affected by some outliers in the detected line segments.

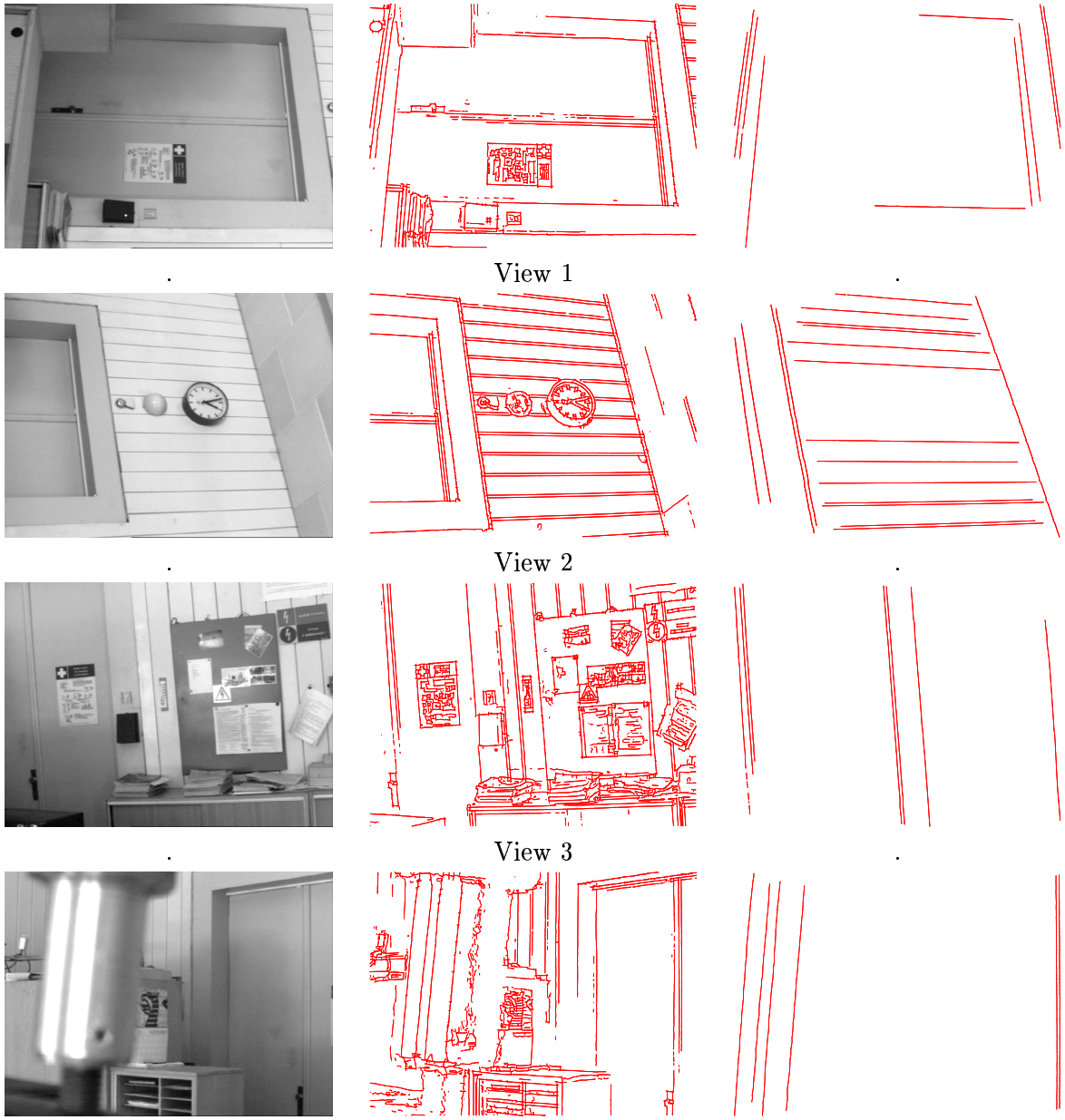
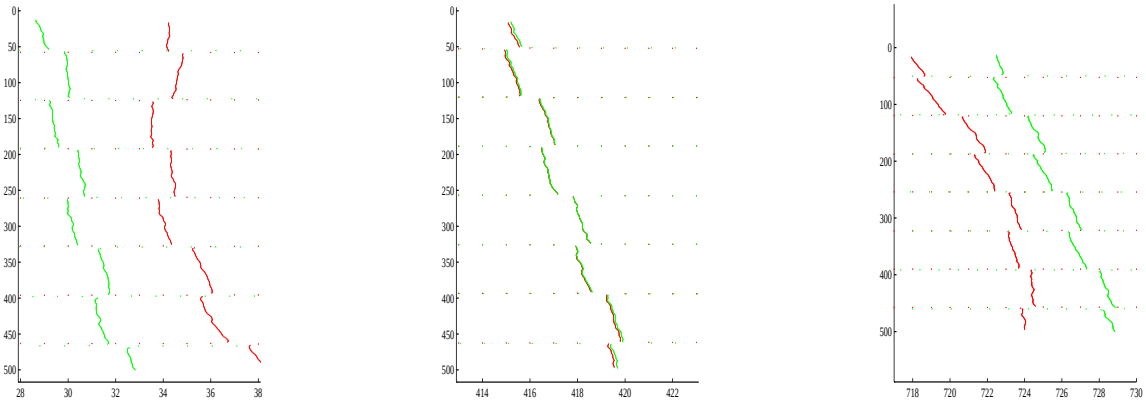


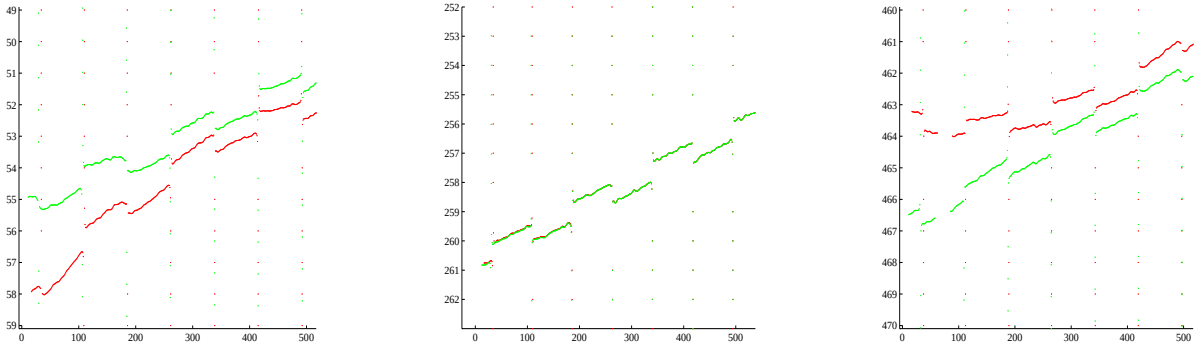
Figure 4.3: Gray level images of the doors (left column), extracted edges (middle column), and selected lines (right column)

.



(a) (b) (c)

Figure 4.4: (a) The most left, (b) the middle, and (c) the most right line before (red) and after (green) the correction of the distortion.



(a) (b) (c)

Figure 4.5: (a) The top, (b) the middle, and (c) the most line before (red) and after (green) the correction of the distortion.

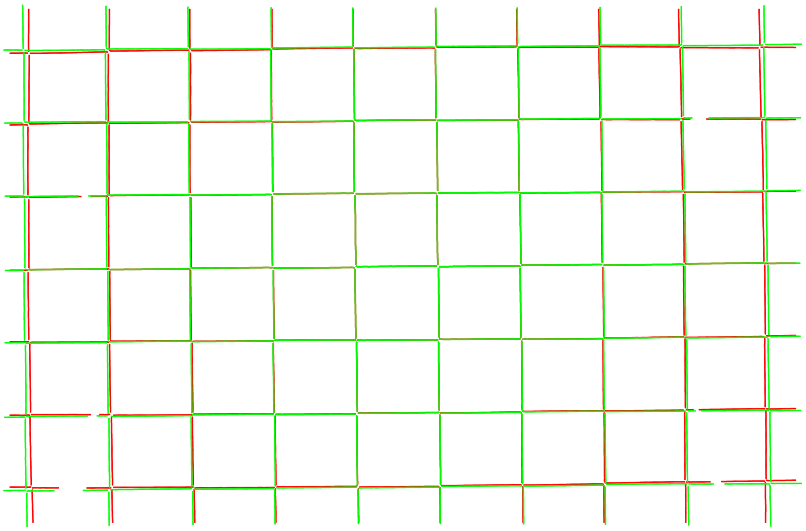


Figure 4.6: Distorted (red) and undistorted (green) Pattern A.

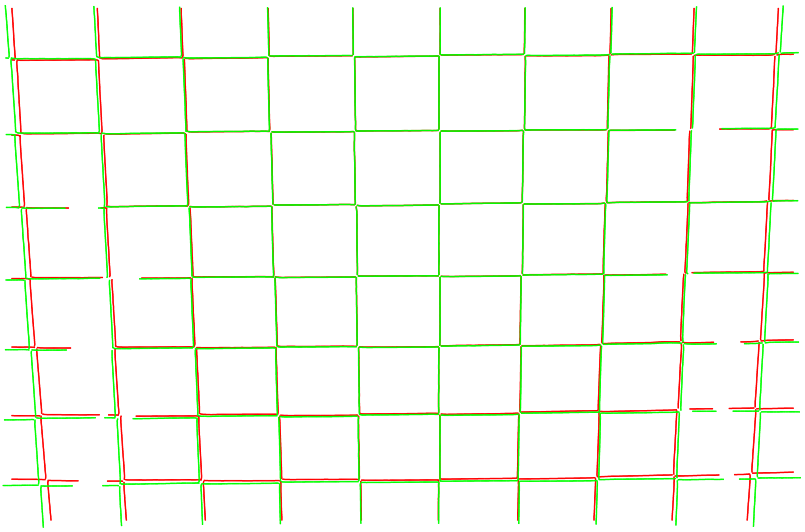


Figure 4.7: Distorted (red) and undistorted (green) Pattern B.

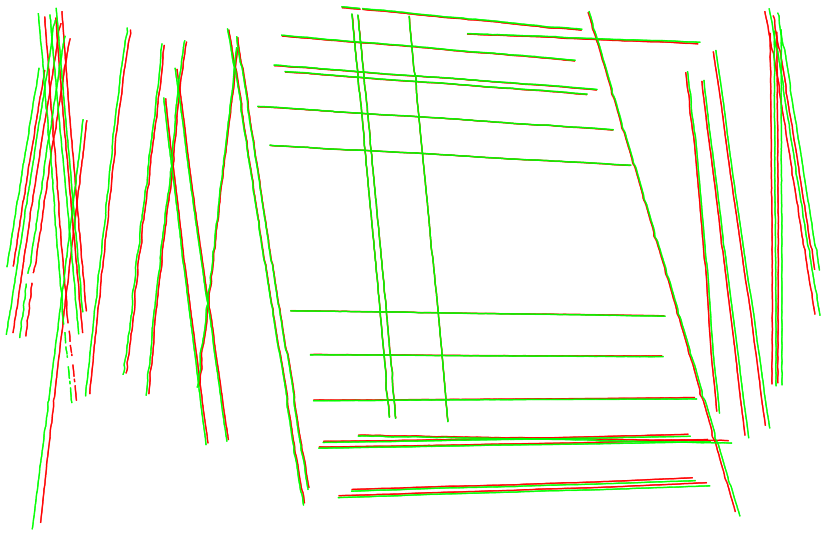


Figure 4.8: Distorted (red) and undistorted (green) linear segments extracted from the scene "doors".

2-nd order distortion model

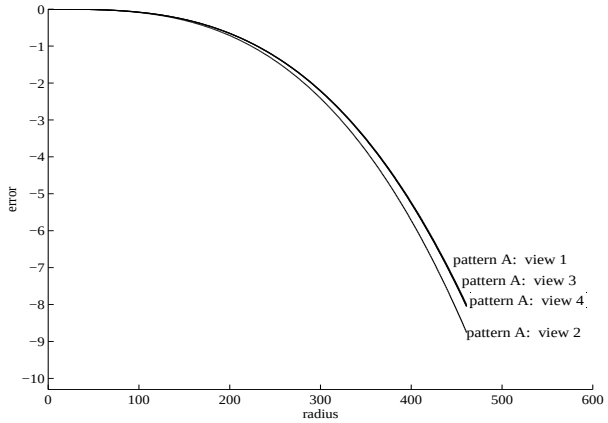
p	e	eu	e/eu	u0	v0	k1
pattern A: view 1	0.347	0.101	3.435	254.6	390.4	−8.2e−008
pattern A: view 2	0.367	0.097	3.794	269.5	394.5	−8.9e−008
pattern A: view 3	0.348	0.100	3.482	254.3	393.3	−8.2e−008
pattern A: view 4	0.348	0.100	3.481	254.6	392.5	−8.2e−008
pattern B: view 1	0.390	0.107	3.635	180.0	397.6	−9.0e−008
pattern B: view 2	0.396	0.108	3.672	201.1	398.6	−1.0e−007
doors: view 1	0.210	0.168	1.245	263.7	434.3	−7.0e−008
doors: view 2	0.202	0.142	1.416	214.1	457.4	−9.8e−008
doors: view 1+2	0.204	0.154	1.329	204.3	453.5	−8.5e−008
doors: view 1+2+3	0.218	0.171	1.275	209.5	397.1	−9.5e−008
loors: view 1+2+3+4	0.232	0.182	1.277	212.7	401.4	−9.7e−008

(a)

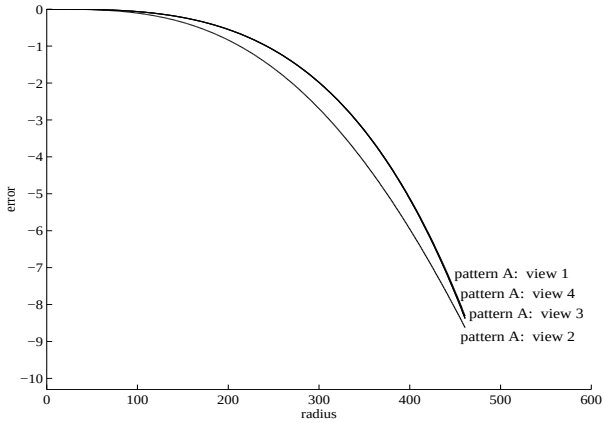
4-th order distortion model

p	e	eu	e/eu	u0	v0	k1	k2
pattern A: view 1	0.347	0.100	3.469	254.2	389.8	−6.5e−008	−9.3e−014
pattern A: view 2	0.367	0.096	3.801	270.4	402.0	−1.1e−007	9.6e−014
pattern A: view 3	0.348	0.099	3.513	253.8	392.7	−6.4e−008	−9.9e−014
pattern A: view 4	0.348	0.099	3.512	254.2	391.9	−6.5e−008	−9.7e−014
pattern B: view 1	0.390	0.104	3.764	176.5	390.5	−1.2e−007	1.4e−013
pattern B: view 2	0.396	0.098	4.031	196.7	389.0	−1.4e−007	1.9e−013
doors: view 1	0.210	0.167	1.256	294.3	390.0	5.1e−008	−4.8e−013
doors: view 2	0.202	0.142	1.420	214.8	455.3	−8.8e−008	−7.7e−014
doors: view 1+2	0.204	0.152	1.345	212.4	469.8	−1.0e−007	8.4e−014
doors: view 1+2+3	0.218	0.172	1.270	184.7	329.5	−1.8e−007	3.4e−013
ors: view 1+2+3+4	0.232	0.183	1.272	185.2	326.8	−1.8e−007	3.4e−013

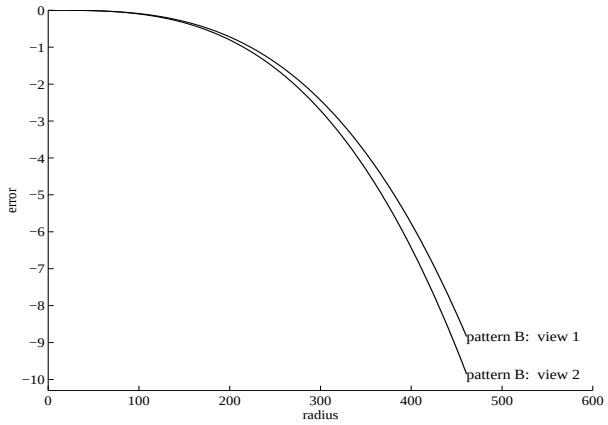
(b)



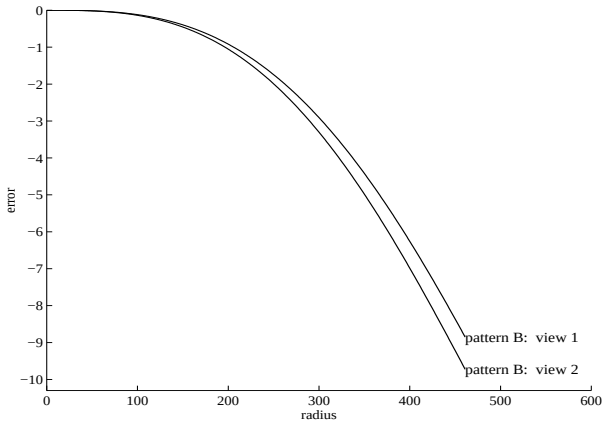
(c)



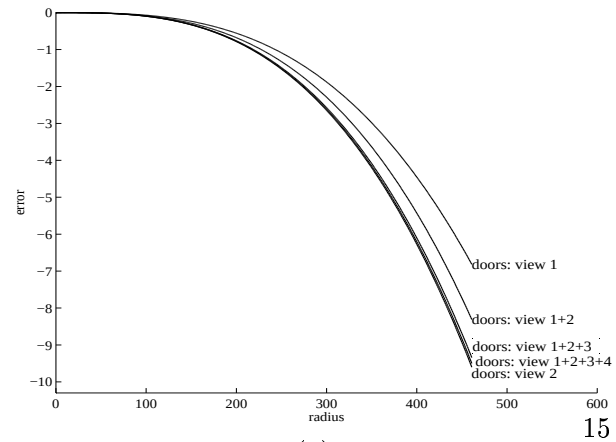
(d)



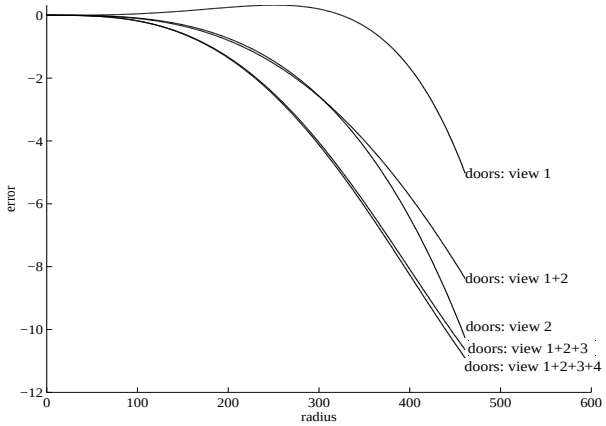
(e)



(f)



(g)



(h)

Figure 4.9: Results for real data. See section 4.2 for explanation.

Chapter 5

Validity of radial distortion model for the scenes with large depth variations

The model of the radial distortion (2.2) assumes that the distortion of a point does not depend on the distance of that point from the camera. The opposite, i.e. having the distortion of the point dependent on its distance from the camera, would require to know the depth of the point in order to correct for its distortion. As the depth of the points can not be obtained from a single view, undoing radial distortion of a single view would be impossible.

It has been reported in [7] (section 3.2.3.3 Finite Object-Distance Lens Calibration) that the distortion of the lens in fact depends on the distance of the points (and also on the distance to which the camera is focused) and is given by the equation

$$D_m = D_0 - mD_\infty, \tag{5.1}$$

where D_0 represents the lens distortion (the distance between a distorted and the correct point) measured at a series of off–axis points in the focal plane while the beam of parallel light from the colimator is incident upon its front element, and D_∞ represents the corresponding distortion quantities while the rear element of the lens is facing the bench colimator. D_m gives then the distortion of the lens at the operational magnification m. The magnification

$$m = \frac{y'}{y}, \tag{5.2}$$

is the ratio of the image size to the object size; here y' and y stand for the size of the image and object respectively. Note, that m is negative due to the convention in optical coordinate system. The magnification can be also expressed [7] (section 3.2.3.3, page 73, equation (3.8)) in terms of a focal length f and the distance L measured from the object to the image plane

$$m = -\frac{1}{f+L}. \tag{5.3}$$

Substituting (5.3) into (5.1) gives the formula for the lens distortion at a distance L

$$D_L = D_0 + \frac{1}{f+L} D_\infty. \tag{5.4}$$

It is assumed that the camera is always focussed to the plane at the distance L.

D.C. Brown has developed a similar method [1] to the one proposed in this work. He also studied the dependency of the radial distortion on the distance of the object and the distance to which the camera is focussed and come to the same conclusion as described in [7] (section

3.2.3.3). He shows that the distortion at a distance L can be obtained as a linear interpolation between distortions at two planes at a distance L_1 and a distance L_2 ([1] ,equation(4))

$$D_L = \alpha_L D_{L_1} + (1 - \alpha_L) D_{L_2}, \tag{5.5}$$

where

$$\alpha_L = \frac{L_2 - L}{L_2 - L_1} \frac{L_1 - f}{L_1 - f}. \tag{5.6}$$

A different situation occurs if the camera is focussed to a fixed distance L and the object is off the focus plane in the distance L'. It is shown in [1] that the distortion $D_{L,L'}$ of the point at the distance L' generated by the lens focussed to the distance L is given by

$$D_{L,L'} = \gamma_{L,L'} D_{L'}, \tag{5.7}$$

where

$$\gamma_{L,L'} = \frac{L - f}{L' - f} \frac{L'}{L}. \tag{5.8}$$

Brown reports the difference between $D_{1.5[m],1.5[m]}$ and $D_{\infty,\infty}$ to be about 60% for the camera Schneider Symmar, $f = 135$ mm.

We have done a similar experiment to the one described by Brown [1]. We have observed planar target with a set of parallel lines placed at four different distances L from the camera (Computar lens, $f = 12$ mm, 3 mm displacement ring between the lens and the camera). For each L, two patterns with horizontal (cca 14) lines and vertical (cca 17) lines were captured and combined in order to get an orthogonal system of lines. The camera was always focused to the target plane by maximizing visual perception of sharpness. Four different targets with varying separations between the lines were used. The separations were chosen to obtain the images of the target almost identical for all four positions. Such an arrangement reduces bias in estimate of the radial distortion due to different (or uneven) distribution of points in the scene. The images were taken by a digital CCD camera with the resolution 768×576 pixels (pixel size $= 10\mu m$). Each image of a target was obtained as an average of 10 consecutive images taken in about 5 seconds. Averaging suppresses the noise caused by light and electronics and improves the stability of line detection in images. Lines were detected by extraction of zero crossing in horizontal resp. vertical directions.

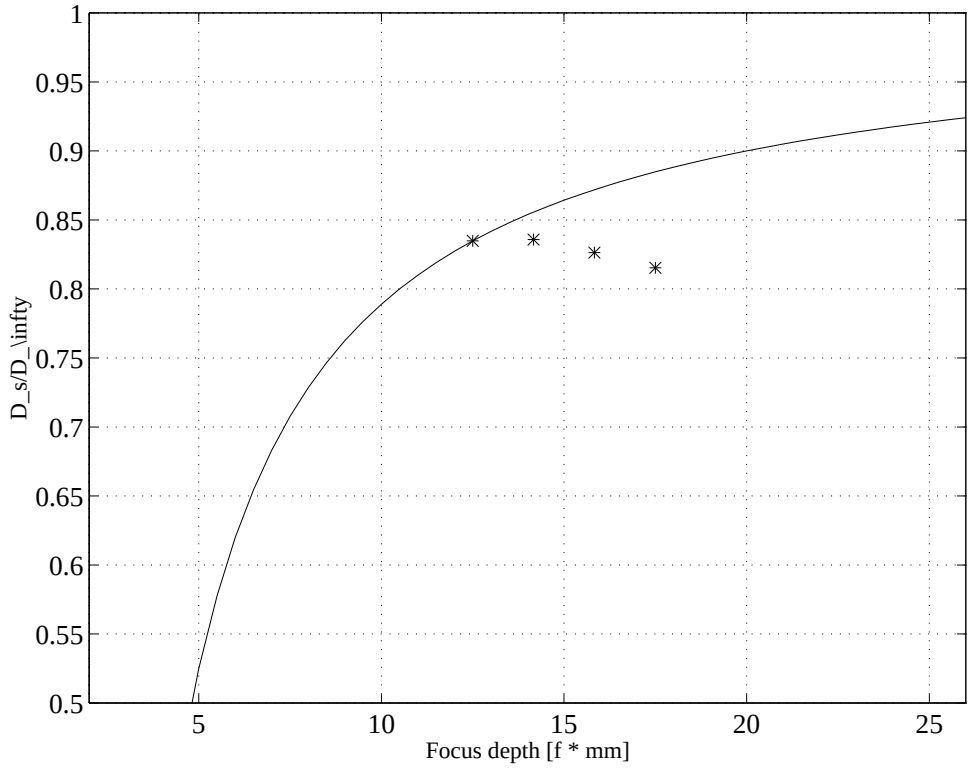
Figure 5.1 compares the results of our measurement with the results reported in [1]. The plots show the ratio of the distortion at the distance L to the distortion at infinity as a function of the distance to the target expressed in the multiples of the focal length f

$$\frac{D_L}{D_\infty} = function(\frac{L}{f}) \tag{5.9}$$

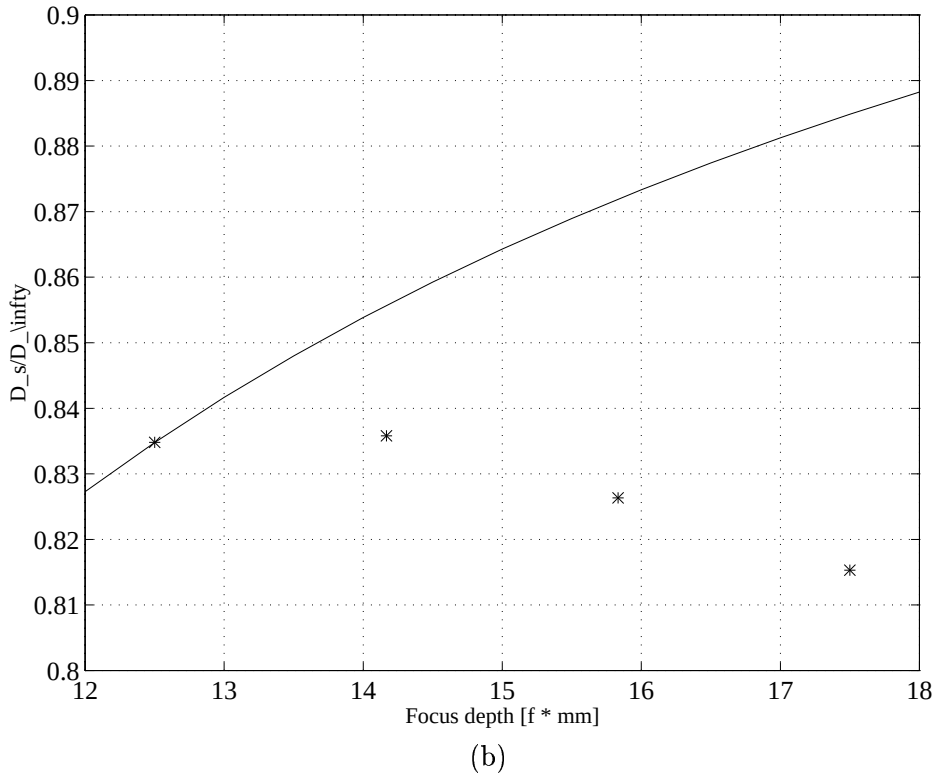
Stars (*) denote data measure by us. Full line shows the function fitted to data from [1]. Our data are scaled so that the first measurement at $L/f = 12.5$ lies on the curve from [1].

The results show that the overall change in our case is about 2% while in the case of data from [1] it is about 5%. We do not know why the distortion of our lens decreases with increasing distance while it increases for data in [1]. Similar results are reported by Stein in [6] who also found that distortion decreases with growing distance.

Overall conclusion could be that the radial distortion certainly changes with the change of distance between the lens and the object but we are not able at the moment judge whether it has to be modeled and when it is important to model it. Further experiments are needed to see how the lenses mostly used in computer vision behave.



(a)



(b)

Figure 5.1: (a) The ratio of the distortion at the distance L to the distortion at infinity as a function of the distance to the target expressed in the multiples of the focal length f. (b) Scaled (a)

Chapter 6

Conclusions

We have proposed two ways how to estimate and compensate for lens radial distortion. For it, knowledge of full scene 3-D structure is not necessary. The first method requires knowledge of lines in the scene, the second method requires knowledge of a coplanar set of point in the scene up to a homographic transformation.

Both methods worked satisfactorily for simulated data. The estimated distortion parameters degraded gracefully with noise added to the measured points, as far as the amount of noise was not comparable with the error caused by the radial distortion.

Only the method estimating from straight lines was tested on real data. The results for calibration grid, from which lines were extracted, were satisfactory. For real environment, it required a certain amount of work to find appropriate lines in this environment which are: (i) long enough, (ii) not distorted due to other reasons than radial distortion.

At the end we investigated the dependence of radial distortion on the depth of observed points. We found that this dependence decreased with the increasing depth, in contrary to the result due to [1]. This important disagreement remains to be explained.

6.1 Other relevant literature

The following works might be relevant to a better understanding of the radial distortion, especially of its behaviour with changing depth: [7] (page 107, section 3.2.3.3), [8], [10], [9], [5], [4].

Appendix A

Programmer's Guide to the RADIAL Toolbox

The above algorithms have been implemented as a MATLAB toolbox. We will briefly describe the functions in this toolbox.

Here is the file `contents.m` of the RADIAL toolbox:

```
%The RADIAL toolbox
%
%  Direct and inverse distortion
%    r2rdist - Lens radial distortion applied to an undistorted image point radius
%    rdist2r - Inverse lens radial distortion applied to a distorted image point radius
%    u2udist - Radial lens distortion function applied for image points
%    udist2u - Inverse radial lens distortion function applied for image points
%
%  Estimating radial distortion
%    rdestim - Estimating radial lens distortion parameters
%    objln   - Objective function for correcting radial lens distortion from lines
%    objho   - Objective function for correcting radial lens distortion from scene points
%              known up to 2D-to-2D homography
%
%  Other
%    simul  - Testing the correction of radial lens distortion on synthetic data
```

Here are help texts of the functions listed in `contents.m`:

```
%r2rdist Lens radial distortion applied to an undistorted image point radius.
%        rd = r2rdist(r0,k,r), where
%          r0 ... Size (1,1). Radius in which rd-r==0.
%          k .... Size (kn,1). Coefficients of the lens distortion polynom.
%          r ... Size (1,N). Radius of the undistorted image point.
%          rd ... Size (1,N). Radius of the distorted image point.
%
%        The model of the lens radial distortion:
%          rd = r*(1 + k(1)*(r^2-r0^2) + k(2)*(r^4-r0^4) + ... )
%
%         See also rdist2r.

%rdist2r Inverse lens radial distortion applied to a distorted image point radius.
%        r = rdist2r(r0,k,rd)
%        Parameters and distortion model are described in r2rdist.
```

```
%
%        rdist2r is the inverse function to r2rdist.
%        Since the inverse cannot be found in a closed form, it is obtained by
%        a rapidly converging iterative algorithm, using the Banach's fixed point theorem.
%
%        See also r2rdist.

%u2udist  Radial lens distortion function applied for image points.
%         u2 = u2udist(p,u1), where
%         u1 ... Size (2,N). Undistorted image points.
%         u2 ... Size (2,N). Distorted image points.
%         p ... Size (?,1). Radial lens distortion parameters. See r2rdist.
%
%         See also UDIST2U, R2RDIST, RDIST2R.

%udist2u  Inverse radial lens distortion function applied for image points.
%         u2 = udist2u(p,u1), where
%         u1 ... Size (2,N). Distorted image points.
%         u2 ... Size (2,N). Undistorted image points.
%         p ... Size (?,1). Radial lens distortion parameters. See r2rdist.
%
%         See also U2UDIST, R2RDIST, RDIST2R.
```

These four functions implement the model described by Eq. 2.3 and its inverse. In fact, the implemented model is Eq. 2.7. If `r0` is set to zero, the simpler model Eq. 2.3 is obtained. The inverse is computed by the algorithm from Section 2.3 with $\lambda = -1$.

```
%rdestim  Estimating radial lens distortion.
%
%       p = rdestim('lines',p0,ud,L) estimates distortion parameters from lines in scene.
%       Parameters:
%         p0 ... The initial estimate of radial lens distortion parameters.
%                p0 = [u0; r0; k], see R2RDIST for details.
%         ud ... Size (2,?). Considered distorted image points of which some are colinear.
%         L ... Size (?,N), sparse. Non-zero entries in a column are indices of points
%               ui which are colinear in the scene. There are N lists of colinear points.
%         p ... Size=size(p0). Estimate of the radial lens distorion parameters.
%       Algorithm:
%         The radial lens distortion parameters p are estimated using the knowledge that
%         some points from ud are colinear in the world. Due to the lens distortion, they
%         are not colinear in the image.
%           The algorithm estimates the parameters p by minimizing residuals of fitting
%         lines to colinear points from ud, after they were "undistorted" by inverse
%         distortion function of p.
%       Notes:
%         Optimization runs always with fixed r0 (=p(3)).
%
%       [p,H] = rdestim('homo',p0,ud,uw)
%       Parameters:
%         p0, p ... As above.
%         ud ... Size (2,M). Projections of coplanar scene points uw to image plane
%                by a camera with radial lens distortion.
%         uw ... Size (2,M). Coplanar scene points known up to a 2D-to-2D homography.
```

```
%         H ... Size (3,3). Homography estimate. It is approx.
%                 ud = UDIST2U(p,p2e(H*e2p(uw))).
%       Algorithm:
%         Using image points ud (projected from the world by a perspective camera
%         with radial distortion) and corresponding world points uw known up to
%         a 2D-to-2D homography H (the world points are coplanar), the function
%         estimates homography H and camera radial distortion parameters p. The
%         estimation is done by gradient descent. The initial value of H is estimated
%         automatically from ud and uw, the initial value of p must be given.
%       Notes:
%         Optimization runs always with fixed r0 (=p(3)).
%
%       See also u2udist, udist2u.
```

This function implements the algorithms from Sections 3.1 and 3.2.

```
%objho Objective function for correcting radial lens distortion from scene points
       known up to 2D-to-2D homography.
%      res = objho(p,ud,uw), where
%        Hp ... Size (?,1). Homography and lens distortion parameters.
%               It is Hp = [H(:); p]. See rdestim.
%        ud, uw ... See rdestim.
%        Res ... Size (?,1). Residuals of 2D-to-2D homography to ud, uw,
%                where ud were first undistorted by p.
```

In the implementation, `Res` is not equal to ρ_{homo} from Eq. 3.5 but rather it is a vector of distances between individual point pairs. The summation is done inside the M-function `leastsq` (Levenberg-Marquardt algorithm).

```
%objln Objective function for correcting radial lens distortion from lines.
%      Res = objln(p,ud,L), where
%        p ... See u2udist.
%        ud, L ... See rdestim.
%        Res ... Size (?,1). Residuals of fitting lines to colinear points,
%                first undistorted by p.
```

In the implementation, `Res` is not equal to ρ_{lines} from Eq. 3.9 but rather it is a vector of distances of individual points to the apprpopriate lines. The summation is done inside the M-function `leastsq` (Levenberg-Marquardt algorithm).

```
%simul Testing the correction of radial lens distortion on synthetic data.
%
%   THE TEST ALGORITHM
%
% The test consists of the following steps:
%
%   1. Generating world points uw.
%      The points in the 3D world are generated. These points are coplanar and their
%      coordinates are expressed in the system of the plane they lie in, eg, their
%      coordinates are 2D. The points form a regular grid. The numbers of grid
%      horizontal and vertical lines are given by GrSize. The range of grid point
%      coordinates is always <0,1>x<0,1>.
%
```

```
%   2. Generating linear-camera image points ul.
%      Projection of the world points by linear camera is done by applying
%      a homography to the world points uw. The homography is such that the resulting
%      image size is approx. ImSize.
%
%   3. Applying radial lens distortion to ul, the result is distorted points ud.
%      See parameters p and the help for U2UDIST for the distortion model.
%
%   4. Adding noise to ud, the result is the points un.
%      The noise portion is given by Sigma. Uniform noise is currently used.
%
%   5. Estimating the lens distortion parameters pu using the points ud.
%      Using rdestim, distortion parameters p are estimated.
%      The initial estimate of p is p0.
%
%   6. Undistorting the distorted points ud by pu, the result is uu.
%      The undistortion function UDIST2U is applied to the NOISELESS distorted points.
%
%   Steps 4, 5, 6 are repeated for various portions of noise Sigma and various trials.
%   The results are visualized.
```

Bibliography

[1] D.C. Brown. Close-range camera calibration. *Jour. Ph. Eng.*, 37(8):855–866, August 1971.

[2] George L. Cain. *Introduction to General Topology.* Addison-Wesley Publishing Company, 1994.

[3] Richard I. Hartley. In defence of the 8-point algorithm. In *Fifth International Conference on Computer Vision*, pages 1064 – 1070. IEEE Copmuter Society Press, 1995.

[4] Trond Melen. *Geometrical Modelling and Calibration of Video Cameras for Underwater Navigation.* PhD thesis, Institutt for teknisk kybernetikk, Norges techniske hogskole, Trondheim, Norway, November 1994. Also technical report ITK No. 94-103-W. Author's e-mail Trond.Melen@itk.unit.no.

[5] B. Prescott and G. F. McLean. Line-based correction of radial lens distortion. *Graphical Models and Image Processing*, 59(1):39–47, January 1997.

[6] Gideon P. Stein. Internal camera calibration using rotation and geometric shapes. Master's thesis, Massatchusetts Institute of Technology, February 1993.

[7] Morris M. Thompson, editor. *Manual of Photogrammetry.* American Society of Photogrammetry, 1968.

[8] Morris M. Thompson, editor. *Manual of Photogrammetry.* Nedra Moskva, 1970. In Russian.

[9] Robert J. Valkenburg. Classification of camera calibrations techniques. Available in CMP Prague, or E-mail to b.valkenburg@irl.cri.nz.

[10] Reg G. Wilson and Steven A. Shafer. What is the center od the image. *Jour. of Opt. Soc. Am.*, 11(11):2946–2955, November 1994.