Improving camera calibration

Applying a least squares method to control point measurement

Peter Johansson

27th May 2005
Master's Thesis in Computing Science, 20 credits
Supervisor at CS-UmU: Niclas Börlin
Examiner: Per Lindström

UmeåUniversity
Department of Computing Science
SE-901 87 UMEÅ
SWEDEN

Abstract

A nonlinear optimization based method for measuring circular control points for use with camera calibration is presented. The displacement of circle centers, caused by projection, is addressed and corrected. Camera calibration is performed taking both radial and tangential lens distortion into account. The results show some improvement with the least squares method over the classical centroid method. The results with and without circle center displacement correction were similar. However, the technique was useful for outlier detection.

Contents

List of Figures

1 Introduction

1.1 Background

Photography is a process to map points in the 3D world to points on a 2D plane. In many applications, like computer vision, map making and medicine, measurements are made in the image and then related to measurements in the 3D world. This technique is called *photogrammetry* and has been around for more than 100 years. To be able to do this correctly an understanding of how the image is taken, a relationship between the world and image, is needed. The process to calculate the parameters that describe this relationship is called *camera calibration*.

The basic idea behind the process of camera calibration is that by taking images of objects with features that are easily identified in the image, a relationship between 3D points and corresponding 2D points in the image can be formed. Using these corresponding points, the parameters of the camera can be calculated.

One of the major problems is to identify and determine positions of image features, something necessary for a correct calibration. For easy identification, squares, crosses or circles are often used as features or *control points* as they are called.

1.2 Aim

The aim of this thesis is to implement and evaluate a new method for determining the position of circular control points used in images for camera calibration. The method derives from taking an optimization approach to the problem by minimizing a mathematical model of a control point over the image of a control point. The method will be evaluated by comparing it to a traditional centroid method and data collected from a commercial software product with camera calibration tools, PhotoModeler[1]. An in depth focus will be placed on the problems occurring when projecting circles.

The report will begin by covering and explaining the major points of the theory needed, followed by explanations of the methods used, then ending with results and conclusions. A basic understanding of calculus and linear algebra is needed to fully profit by this thesis.

1.3 Related work

A lot of papers and books have been written concerning camera calibration and its applications [1] [7] [4] [3] but not as much work has been done specifically about determining control point positions. In case of circular control points almost always a centroid calculation is used.

[1]For more information about PhotoModeler visit *www.photomodeler.com*

2 Theory

2.1 Projective geometry

Projective geometry is a more general geometry than ordinary euclidean geometry. It allows for parallel lines and planes to intersect in an *ideal* point or line at infinity. To be able to denote this, homogeneous coordinates are used. Using homogeneous coordinates all points in 2D are denoted as a vector of length three. A homogeneous point (x, y, w) represents the euclidean point $(x/w, y/w)$. If $w = 0$ the point is at infinity.

As the exception of parallelity is removed the concept of *duality* can be introduced. Since two points always form a line and two lines always intersect in a point, all that applies to points applies to lines. For example by using homogeneous coordinates the line, $\mathbf{l}$, created by two points, $\mathbf{x}_1$, $\mathbf{x}_2$, is calculated as $\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$. And since points are the duality of lines, a point, $\mathbf{x}$, can be calculated as the intersection of two lines, $\mathbf{l}_1$, $\mathbf{l}_2$ according to $\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$.

2.2 2D homography

A 2D homography, $\mathbf{H}$, is a projective transformation between a set of points, $\mathbf{x}_i \in \mathbb{P}^2$, and a set of points, $\mathbf{x}'_i \in \mathbb{P}^2$, such that $\mathbf{H}\mathbf{x}_i = \mathbf{x}'_i$. In other words $\mathbf{H}$ is a 3×3 matrix which transforms $\mathbf{x}_i$ to $\mathbf{x}'_i$.

Since $\mathbf{H}$ has 8 degrees of freedom excluding scale, at least 4 point pairs are needed to calculate it. The calculation is done by taking into fact that $\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = 0$. By writing

$$\mathbf{x}'_i = \begin{bmatrix} x'_i \\ y'_i \\ w'_i \end{bmatrix}, \mathbf{H}\mathbf{x}_i = \begin{bmatrix} \mathbf{h}_1^T \mathbf{x}_i \\ \mathbf{h}_2^T \mathbf{x}_i \\ \mathbf{h}_3^T \mathbf{x}_i \end{bmatrix}$$

where $\mathbf{h}_1^T, \mathbf{h}_2^T, \mathbf{h}_3^T$ are the row vectors of $\mathbf{H}$, $\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = 0$ can be written as

$$\begin{bmatrix} 0_3^T & -w'_i\mathbf{x}_i^T & y'_i\mathbf{x}_i^T \\ w'_i\mathbf{x}_i^T & 0_3^T & -u'_i\mathbf{x}_i^T \\ -v'_i\mathbf{x}_i^T & u'_i\mathbf{x}_i^T & 0_3^T \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} = 0 \tag{1}$$

The third row is linearly dependent on the other two and can be removed. Stacking the two rows for each point results in a $2n \times 9$ matrix $\mathbf{A}$.

$$\mathbf{A} = \begin{bmatrix} 0_3^T & -w'_1\mathbf{x}_1^T & v'_1\mathbf{x}_1^T \\ w'_1\mathbf{x}_1^T & 0_3^T & -u'_1\mathbf{x}_1^T \\ \vdots & \vdots & \vdots \\ 0_3^T & -w'_n\mathbf{x}_n^T & v'_n\mathbf{x}_n^T \\ w'_n\mathbf{x}_n^T & 0_3^T & -u'_n\mathbf{x}_n^T \end{bmatrix}$$

Then $\mathbf{Ah} = 0$ can be solved, ignoring the obvious solution $h = 0$.

2.3 Camera model

The theory in this chapter is mainly from [4]. A camera is generally speaking a device for mapping between the 3D world and a 2D image. When modeling a camera there are two major classes, those models with finite centers and those models with centers at infinity.

2.3.1 Pinhole camera

The most basic representation of a finite camera is the pinhole camera. This model describes the camera in a way that works well with digital cameras and is therefor used here. An illustration of a pinhole camera can be seen in figure 1.

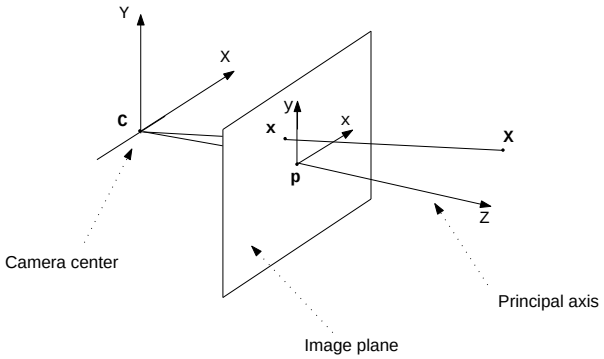


Figure 1: Illustration of the pinhole camera model

The center of the projection, $\mathbf{c}$, is called the *camera center*. The line from the camera center perpendicular to the image plane is called the *principal axis*, and the point, $\mathbf{p}$, where it intersects the image plane is called the *principal point*. A 3D point $\mathbf{X} = (x, y, z)^T$ is mapped to a 2D point, $\mathbf{x}$, on the image plane where the line between $\mathbf{X}$ and the camera center, $\mathbf{c}$, intersects the image plane. The distance between the camera center and the principal point is called the *focal length* and is denoted f [4]. Using homogeneous coordinates a simple camera equation can now be written as

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

where the camera is centered at $(0, 0, 0)^T$.

2.3.2 Principal point offset

The above equation assumes that the principal point is at the origin, $(0, 0)^T$, in the image plane. This is not generally true so an offset is introduced which puts the principal point at $(p_x, p_y)^T$.

$$\begin{bmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3}$$

The part of the matrix which include the internal parameters is called the calibration matrix and is henceforth denoted as **K**. In this instance

$$\mathbf{K} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{4}$$

The other part of the matrix is the camera center which in this instance is $(0, 0, 0)^T$.

2.3.3 Digital cameras

The previous equations assume that the image coordinate system is equally scaled in both axis directions. In case of the digital cameras which use a CCD display consideration has to be taken to the fact that the pixels might not be square. This would lead to axes which are not equally scaled. To solve this m_x and m_y are introduced which is the distance per pixel in both directions. This gives a calibration matrix K for digital cameras which is

$$\mathbf{K} = \begin{bmatrix} a_x & & x_0 \\ & a_y & y_0 \\ & & 1 \end{bmatrix} \tag{5}$$

where $a_x = fm_x$, $a_y = fm_y$, $x_0 = p_x m_x$ and $y_0 = p_y m_y$.

2.3.4 General finite camera

A final parameter is now added to make the model more general. The skew parameter s is there in case the image plane axes are not perpendicular. This will almost never happen so s will almost always be 0.

$$\mathbf{K} = \begin{pmatrix} a_x & s & x_0 \\ 0 & a_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

The projection matrix, $\mathbf{P}$, can now be created according to

$$\mathbf{P} = \mathbf{KR}[\mathbf{I}|-\mathbf{c}] \tag{7}$$

where $\mathbf{c}$ is the camera position in world coordinates and $\mathbf{R}$ is a rotation matrix specifying the cameras rotation. All in all the camera model has 11 degrees of freedom.

So the depiction of a world point $\mathbf{X}$ to a image point $\mathbf{x}$ is done according to

$$\mathbf{x} = \mathbf{PX} \tag{8}$$

2.3.5 Lens distortion

The lens is a pivotal part of the camera. The purpose of the lens is to focus the light from one point in the world to one point in the image. To achieve focus without the use of a lens would require a pinhole type camera with an infinitely small hole. Also it would have to remain open an infinitely long time to obtain enough light.

The disadvantage of using lenses is that as they refract light the position of the projected point on the image plane is altered, as seen in figure 2.

The distortion of a point can be expressed as a symmetric distortion, *radial*, and an asymmetric distortion, *tangential* [7].

All lenses produce a radial distortion, resulting from the effects discussed above, which can be modeled as

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = L(\nabla x, \nabla y) \begin{bmatrix} \nabla x \\ \nabla y \end{bmatrix} \tag{9}$$

where $(\nabla x, \nabla y)^T = (\tilde{x} - x_p, \tilde{y} - y_p)^T$ is the undistorted point, $(\tilde{x}, \tilde{y})^T$, relative to the principal point, $(x_p, y_p)^T$, and $L(\nabla x, \nabla y) = K_1 r^2 + K_2 r^4 + ...$ is a function where $r = \sqrt{\nabla x^2 + \nabla y^2}$. For an illustration of radial lens distortion, see figure 3.

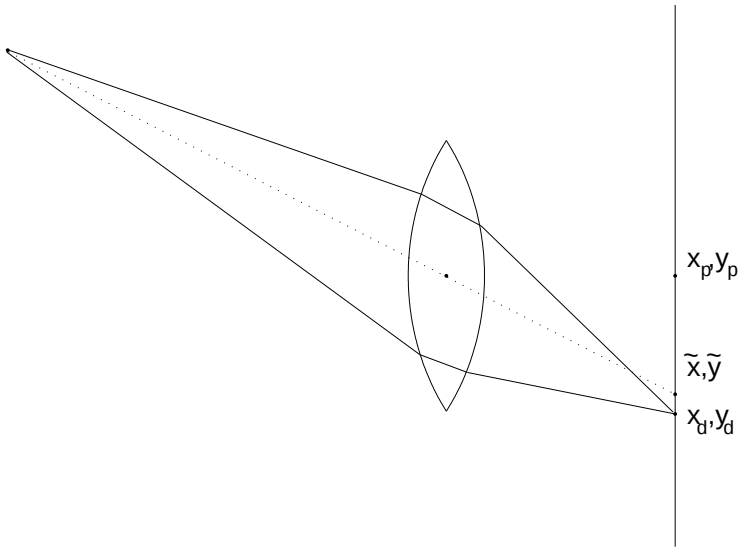


Figure 2: Illustration of the lens distortion where (x_p, y_p) is the principal point, $(\tilde{x}, \tilde{y})$ is the undistorted projected point and (x_d, y_d) is the distorted projected point

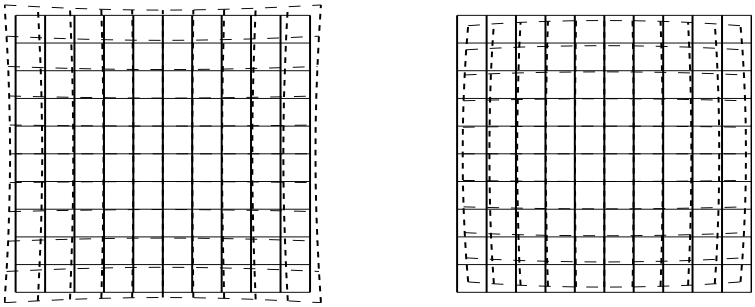


Figure 3: Effects of radial lens distortion. The first distortion is called *Pin cushion* and the second *Barrel*.

Tangential distortion is the result of non-perfect centering of the lens components and other manufacturing defects. It is often very small and can in most cases be disregarded. The effect is that the symmetric center of the distortion is no longer the principal point. This leads to an asymmetric distortion, see figure 4, and can be modeled as

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} 2P_1\nabla x\nabla y + P_2\left(r^2 + 2\nabla x^2\right) \\ 2P_2\nabla x\nabla y + P_1\left(r^2 + 2\nabla y^2\right) \end{bmatrix} \tag{10}$$

where P_1 and P_2 are the distortion parameters.

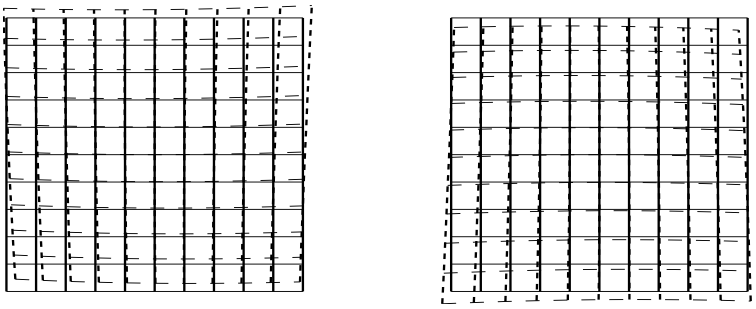


Figure 4: Effects of tangential lens distortion

The combined distortion, see figure 5, can then be modeled as

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \end{bmatrix} \tag{11}$$

and the distorted point is then calculated as undistorted point plus distortion.

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{12}$$

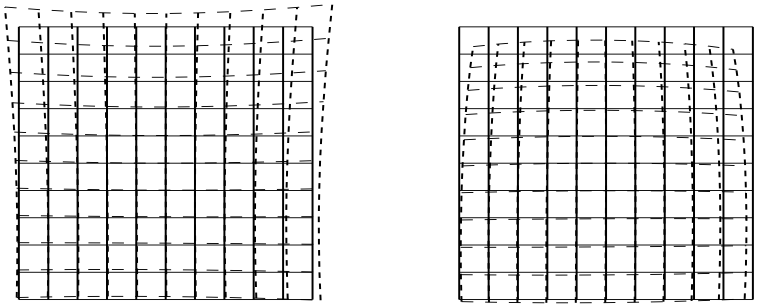
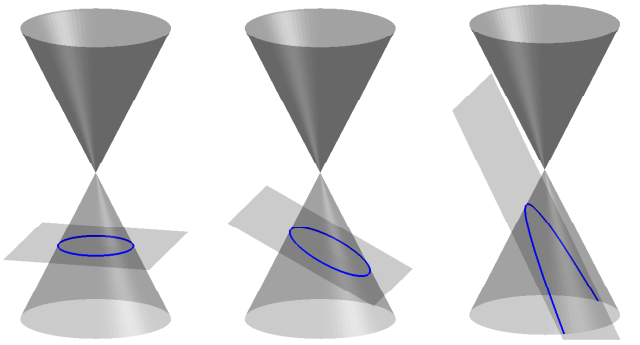


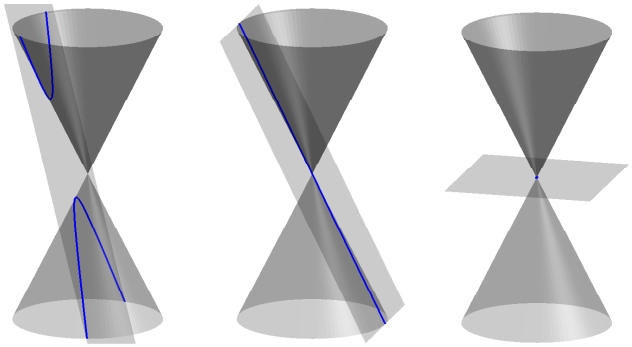
Figure 5: Effects of combined radial and tangential lens distortion

2.4 Conics

The theory in this chapter is mainly from [8]. There are several types of conics or conic sections, circle, ellipse, parabola, hyperbola, point and line. They are called conics because they can be seen as slices of a cone. See figure 6.



(a) Circle (b) Ellipse (c) Parabola



(d) Hyperbola (e) Line (f) Point

Figure 6: Illustration of the different conic sections

The general equation of a conic is

$$Ax^2 + By^2 + Cxy + Dx + Ey + F = 0 \tag{13}$$

and when written on matrix form and using homogeneous coordinates

$$\mathbf{p}^T\mathbf{E}\mathbf{p} = 0 \tag{14}$$

$$\mathbf{E} = \begin{pmatrix} A & C/2 & D/2 \\ C/2 & B & E/2 \\ D/2 & E/2 & F \end{pmatrix}$$

$$\mathbf{p} = \begin{pmatrix} y \\ x \\ 1 \end{pmatrix}$$

which is true for all points $\mathbf{p}$ on the conic.

Even though this is a convenient way of writing a conic the parameters $A, ..., F$ do not say much about the shape and size of the conic. The parameters D, E, F can be eliminated by writing it on this form

$$A(x - x_c)^2 + B(y - yc)^2 + C(x - x_c)(y - y_c) = 1 \tag{15}$$

and on matrix form

$$\mathbf{p}^T \mathbf{E}_2 \mathbf{p} = 0 \tag{16}$$

$$\mathbf{E}_2 = \begin{pmatrix} A & C/2 & 0 \\ C/2 & B & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\mathbf{p} = \begin{pmatrix} x - x_c \\ y - y_c \\ 1 \end{pmatrix}$$

where x_c, y_c represent the center of the conic. To remove the crossproduct term which represents the rotation of the conic a standard rotation matrix $\mathbf{R}$ can be used,

$$\mathbf{E}_3 = \mathbf{R}^T \begin{pmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & -1 \end{pmatrix} \mathbf{R} \tag{17}$$

where $\mathbf{A}$ and $\mathbf{B}$ now directly relate to the major and minor axis of the conic.

In the same way as points are the duality of lines, conics are the duality of line-conics,

$$\mathbf{l}^T \mathbf{E}^{-1} \mathbf{l} = 0 \tag{18}$$

where $\mathbf{l}$ represent all lines tangent to the conic.

2.4.1 Projection of circles

Perspective projection, which the camera model is, does not preserve the shape of the object transformed. Objects in two and three dimensions are distorted if they are not coplanar to the image plane.

In the case of the circle the perspective projection will be a conic, more specifically an ellipse or if coplanar a circle again. To see this a circle lying in the Z-plane with a radius of R can be defined as

$$C(X,Y) = X^2 + Y^2 - R^2 \tag{19}$$

The image points are calculated as follows

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \tag{20}$$

Solving for x and y yields

$$x = \frac{p_{11}X + p_{12}Y + p_{14}}{p_{31}X + p_{32}Y + p_{34}}$$

$$y = \frac{p_{21}X + p_{22}Y + p_{24}}{p_{31}X + p_{32}Y + p_{34}}$$

which can be rewritten as

$$X = Ax + By + C$$

$$Y = Dx + Ey + F$$

where A, B, C, D, E, F are constants consisting of elements of $\mathbf{P}$. Substituting X, Y into C gives

$$C(X,Y) = C(Ax+By+C, Dx+Ey+F) = (Ax+By+C)^2+(Dx+Ey+F)^2-R^2 \tag{21}$$

which is a general conic [3].

As the projection is not shape preserving the circle center will not coincide with the ellipse center [3]. Se figure 7.

It is not possible to calculate the projected circle centers from one ellipse. However if we have two ellipses which are the result of perspective projection of two equal sized circles their is a solution. See figure 8.

Its obvious that the two non crossing tangents touch the circles on points that if connected create lines that pass over the centers, as seen in figure 9.

As can be seen in figure 10 these four tangent points when projected will be four tangent points on the ellipses. Because lines are projected to lines the projected circle centers will be on the lines between these tangent points.

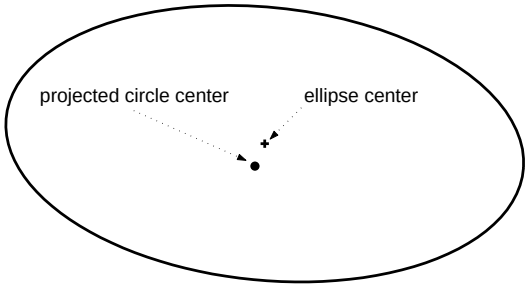
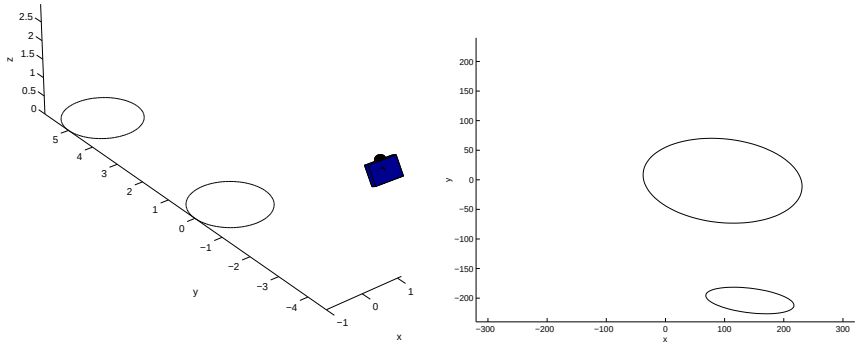


Figure 7: Effect on a circles center when the circle projected



(a) 3D model (b) Resulting image

Figure 8: Projection of two equally sized circles

Figure 9: Common tangents to two equally sized circles

So the problem of finding the project circle centers in the ellipses is reduced to finding the common tangents to the two ellipses. Formulating this problem is quite easy. There are four tangent lines so four equations are needed. The first two are the two ellipse equations. The third one guarantees that we are looking for points with the same slope, the last one that the slope is the slope of the line connecting these two points.

Just as easy as it is to formulate as hard is it to solve. An attempt to solve it using *maple* produced after just a couple of steps equations several pages long.

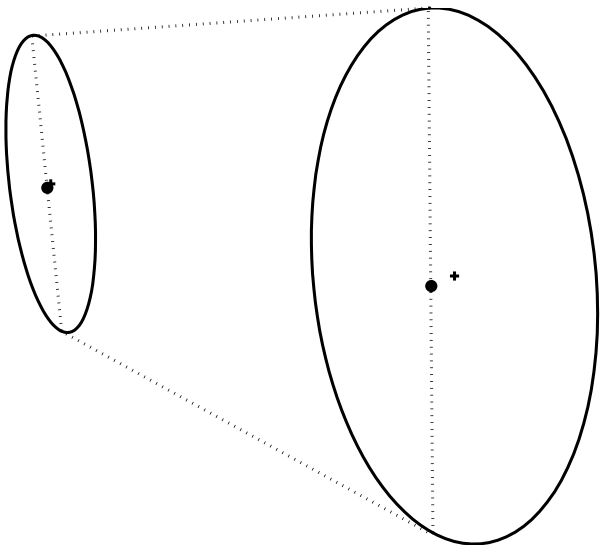


Figure 10: Common tangents to two ellipses resulting from projection of two equally sized circles

Another approach to solve the problem is found by going back to the conic definitions. As stated previously the duality of conics are line-conics,

$$\mathbf{l}^T\mathbf{E}^{-1}\mathbf{l} = 0 \tag{22}$$

where $\mathbf{l}$ represent all lines tangent to the conic.

If $\mathbf{A}$ and $\mathbf{B}$ are the two conics, a linear combination can be formed with their respective line-conics.

$$(t\mathbf{A}^{-1} + (1-t)\mathbf{B}^{-1}) \tag{23}$$

where $t \in \Re$. These combinations will have the same common tangents as the original two conics, see figure 11.

Three of these combinations will be line-conics that will not have full rank. In other words they are defect, having two tangents in the same point which have different slopes. These points are the points where the tangents intersect, see figure 12. The defect conics are in fact only defined in these points.

From the intersection points the tangent points can be calculated easily. As seen in figure 13 the intersection of the line between the tangent points and the line between the two points defining the third degenerate conic, marked with dots in figure 12, is the projected circle center.

So the problem is reduced to finding the three different t which create the degenerate line-conics which satisfy

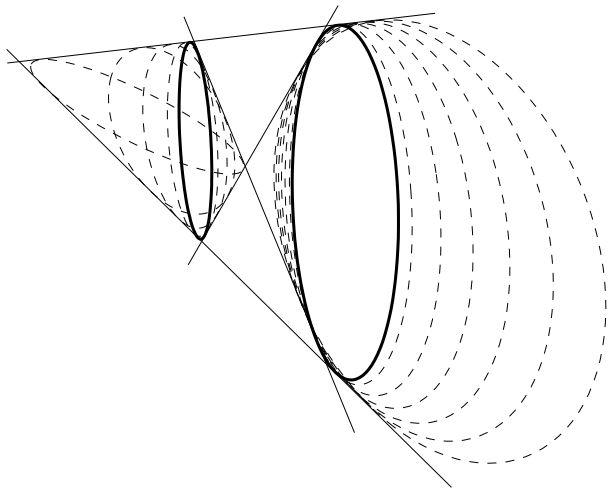


Figure 11: Illustration of some of the conics created by a linear combinations of two other conics

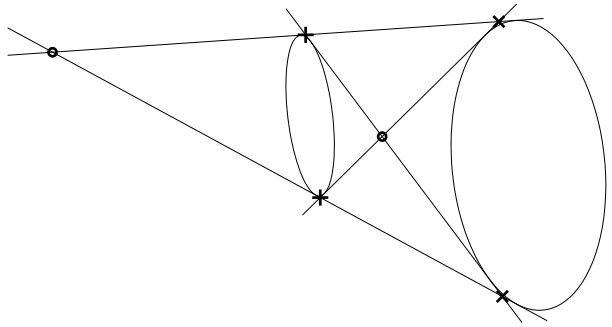


Figure 12: The four common tangents to two ellipses

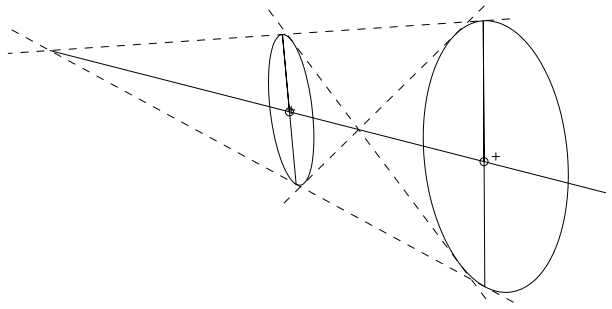


Figure 13: Projected circle center is the intersection between the line connecting the tangent points and the line between the points on the third degenerate conic

$$\mathbf{l}^T(t\mathbf{C} + (1-t)\mathbf{D})\mathbf{l} = 0 \tag{24}$$

where $\mathbf{C} = \mathbf{A}^{-1}$ and $\mathbf{D} = \mathbf{B}^{-1}$. l represent the common tangents of $\mathbf{A}$ and $\mathbf{B}$. To ensure they are degenerate they also have to satisfy

$$det(t\mathbf{C} + (1-t)\mathbf{D}) = 0 \qquad \Longrightarrow$$

$$det\left(\begin{bmatrix} tc_{11} - td_{11} + d_{11} & tc_{12} - td_{12} + d_{12} & tc_{13} - td_{13} + d_{13} \\ tc_{21} - td_{21} + d_{21} & tc_{22} - td_{22} + d_{22} & tc_{23} - td_{23} + d_{23} \\ tc_{31} - td_{31} + d_{31} & tc_{32} - td_{32} + d_{32} & tc_{33} - td_{33} + d_{33} \end{bmatrix}\right) = 0 \tag{25}$$

Evaluating this leads to an equation on the form

$$c_3t^3 + c_2t^2 + c_1t + c_0 = 0 \tag{26}$$

where c_0, c_1, c_2, c_3 are constants created from elements of C and D. An equation on this form can then easily be solved using a numerical method. Each of the three t leads to a degenerate conic $\mathbf{E}_i = (t_i\mathbf{C} + (1-t_i)\mathbf{D})$. To extract the two points, where each degenerate conic is defined, they are factorized into two linear factors

$$\mathbf{l}^T\mathbf{E}_i\mathbf{l} = \mathbf{l}^T(\mathbf{x}\mathbf{y}^T + \mathbf{y}\mathbf{x}^T)\mathbf{l} = 0 \tag{27}$$

which is only true if $\mathbf{x}$ and $\mathbf{y}$ are the two points where $\mathbf{E}_i$ is defined.

2.5 Nonlinear optimization

The theory in this chapter is mainly from [5]. A minimization problem can be formulated as

$$\min_{\mathbf{x}} f(\mathbf{x}), \tag{28}$$

where $\mathbf{x} \in \Re^n$ is a parameter vector and $f(\mathbf{x}) \in \Re$.

Looking at the Taylor expansion of the function f around a start value $\mathbf{x}$ we get the following definitions

$$f(\mathbf{x} + \mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T\nabla^2 f(\mathbf{x})\mathbf{p} + ... \tag{29}$$

where $\nabla f(\mathbf{x})$ is the gradient and $\nabla^2 f(\mathbf{x})$ is the Hessian defined as

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

and

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial^2 x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

2.5.1 The Newton method

Using the Newton method the function f is approximated at the solution of the minimization problem, $\mathbf{x}_*$, using the first two steps of the Taylor expansion [5]. The approximation is called $Q(\mathbf{p}_k)$ as its a quadratic function.

$$f(\mathbf{x}_*) \approx Q(\mathbf{p}_k) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p}_k + \frac{1}{2}\mathbf{p}_k^T \nabla^2 f(\mathbf{x}_k)\mathbf{p}_k \tag{30}$$

For a solution of the minimization problem, $\mathbf{x}_*$, to be a solution two conditions have to be fulfilled.

The *first order necessary condition* is that the gradient is zero

$$\nabla f(\mathbf{x}_*) = 0. \tag{31}$$

The *second order necessary condition* is that the Hessian $\nabla^2 f(\mathbf{x}_*)$ is positive semi definite. By requiring this we guarantee that

$$\frac{1}{2}\mathbf{p}^T \nabla^2 f(\mathbf{x}_*)\mathbf{p} > 0 \text{ or } f(\mathbf{x}_* + \mathbf{p}) > f(\mathbf{x}_*) \quad \forall \mathbf{p} \neq 0 \tag{32}$$

which means that $\mathbf{x}_*$ is a strict local minimizer of f and not a maximizer.

By using the *first order necessary condition* we get

$$\nabla f(\mathbf{x}_*) \approx \nabla Q(\mathbf{p}_k) = \nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)\mathbf{p}_k = 0 \tag{33}$$

Solving for $\mathbf{p}_k$

$$\mathbf{p}_k = -\nabla^2 f(\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k) = 0 \tag{34}$$

which is called the *search direction*. As $Q(\mathbf{p}_k)$ is only an approximation the *search direction* $\mathbf{p}_k$ is also only a first step, meaning $\mathbf{x}_k + \mathbf{p}_k \approx \mathbf{x}_*$.

As $\mathbf{x}_k + \mathbf{p}_k$ is closer to $\mathbf{x}_*$ then $\mathbf{x}_k$ according to the *second order necessary condition* we update, $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$, and repeat the process.

2.5.2 Nonlinear least squares optimization

In this special case a model function $G(\mathbf{x})$ is fitted to m data points in a vector, $\mathbf{d}$, in a way that minimizes the squared distance between them. The problem can be formulated as

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} \frac{1}{2} \sum_{i=1}^{m} (g_i(\mathbf{x}) - d_i)^2 = \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{F}(\mathbf{x})\|^2 = \min_{\mathbf{x}} \frac{1}{2} \mathbf{F}(\mathbf{x})^T \mathbf{F}(\mathbf{x}) \tag{35}$$

where the residual function, $\mathbf{F}(\mathbf{x})$, is formulated as

$$\mathbf{F}(\mathbf{x}) = \mathbf{G}(\mathbf{x}) - \mathbf{d} \tag{36}$$

and the gradient of the function $f(\mathbf{x})$ is

$$\nabla f(\mathbf{x}) = \nabla \mathbf{F}(\mathbf{x}) \mathbf{F}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{F}(\mathbf{x}) \tag{37}$$

where $\mathbf{J}(\mathbf{x})$ is called the *Jacobian* and is defined as

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

The Hessian of the function $f(\mathbf{x})$ is

$$\nabla^2 f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + \sum_{i=1}^{m} f_i(\mathbf{x}) \nabla^2 f_i(\mathbf{x}) \tag{38}$$

Calculating the Hessian for each step of the iteration would be costly. Therefor the *Gauss-Newton* method approximates the Hessian by just using the $\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x})$ part [5]. Since the Jacobian is already needed in the gradient calculation, this will not be costly. Given this approximation the search direction $\mathbf{p}_k$ is calculated as

$$\mathbf{p}_k = -\left(\mathbf{J}\left(\mathbf{x}_k\right)^T\mathbf{J}\left(\mathbf{x}_k\right)\right)^{-1}\mathbf{J}\left(\mathbf{x}_k\right)^T\mathbf{F}\left(\mathbf{x}_k\right) \tag{39}$$

2.6 Camera calibration

The camera calibration process is the process where the camera parameters are calculated, given a number of world positioned points and their corresponding image points [1] [4].

2.6.1 Calibration objects

There are many different kinds of objects used for camera calibration. Since the idea is to find information in the image for which actual information is known, what is needed are things that are easily identified. Often markers like circles or squares are placed on objects with known properties, for instance a box or a piece of paper [7]. In this way the position in the world of all the markers are easily described.

2.6.2 DLT method

The DLT, or *Direct linear transformation*, method is a non-iterative method for solving the calibration problem [4].

What is given is a number of 2D to 3D point correspondences,

$$\mathbf{x}_i = \mathbf{P}\mathbf{X}_i \tag{40}$$

where $\mathbf{x}_i$ are the 2D points, $\mathbf{X}_i$ are the 3D points and $\mathbf{P}$ is what is to be solved. Since the points are expressed in homogeneous coordinates the vectors $\mathbf{x}_i$ and $\mathbf{P}\mathbf{X}_i$ do not have to have the same length, just the same direction. In other words

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = 0 \tag{41}$$

By writing,

$$\mathbf{x}_i = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix}, \mathbf{P}\mathbf{X}_i = \begin{bmatrix} \mathbf{p}_1^T\mathbf{X}_i \\ \mathbf{p}_2^T\mathbf{X}_i \\ \mathbf{p}_3^T\mathbf{X}_i \end{bmatrix}$$

where $\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T$ are the row vectors of $\mathbf{P}$, the following arises

$$\begin{bmatrix} 0_4^T & -w_i\mathbf{X}_i^T & v_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & 0_4^T & -u_i\mathbf{X}_i^T \\ -v_i\mathbf{X}_i^T & u_i\mathbf{X}_i^T & 0_4^T \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = 0 \tag{42}$$

The third row is linearly dependent on the other two and can be removed. Stacking the two rows for each point results in a $2n \times 12$ matrix $\mathbf{A}$.

$$\mathbf{A} = \begin{bmatrix} 0_4^T & -w_1\mathbf{X}_1^T & v_1\mathbf{X}_1^T \\ w_1\mathbf{X}_1^T & 0_4^T & -u_1\mathbf{X}_1^T \\ \vdots & \vdots & \vdots \\ 0_4^T & -w_n\mathbf{X}_n^T & v_n\mathbf{X}_n^T \\ w_n\mathbf{X}_n^T & 0_4^T & -u_n\mathbf{X}_n^T \end{bmatrix}$$

Since $\mathbf{P}$ has 12 unknowns and 11 degrees of freedom when ignoring scale, at least 11 equations are needed. So by using $5\frac{1}{2}$ point pairs an exact solution can be obtained by solving $\mathbf{Ap} = 0$, where $\mathbf{p}$ is a vector consisting from the row vectors of $\mathbf{P}$.

If more point pairs are available, and they are not exact, an exact solution will not exist. To obtain a solution, $\|\mathbf{Ap}\|$ is minimized under the constraint $\|\mathbf{p}\| = 1$, this is to avoid the obvious solution $\mathbf{p} = 0$. The solution vector, $\mathbf{p}$, is the singular vector corresponding to the smallest singular value of $\mathbf{A}$. The camera matrix $\mathbf{P}$ is then constructed from the elements of $\mathbf{p}$.

2.6.3 Least squares method

One of the problems with the DLT method is that it does not incorporate lens distortion. To solve this a least squares method can be used, where the error between the image points and the projected world points is minimized,

$$\min_{\mathbf{P}} \sum d\left(\mathbf{x}_i - \mathbf{P}\mathbf{X}_i\right)^2 \tag{43}$$

where d is the euclidean distance between the measured image points, $\mathbf{x}_i$, and the know world points, $\mathbf{X}_i$.

Residual and Jacobian functions are created which incorporate lens distortion and the problem is solved using the Gauss Newton method.

3 Implementation

A calibration object which consists of a paper with 100 control points arranged in a 10x10 grid has been used. See figure 14.

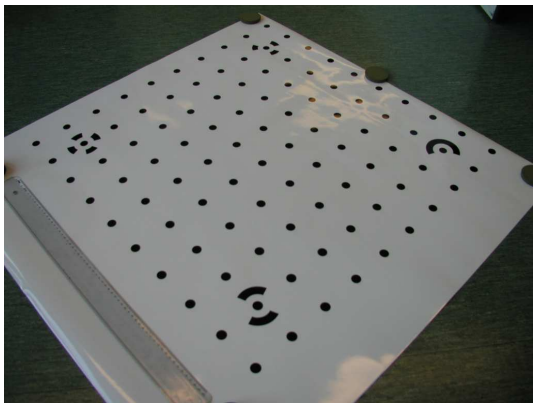


Figure 14: Example of a calibration object where four of the control points are coded.

3.1 Homography estimation

As seen in figure 14 four of the control points are coded. When a new image is presented the user is asked to mark the center of these four coded control points. By creating a virtual square grid, $\mathbf{x}_i$, a homography, $\mathbf{H}$, can be calculated by matching the four virtual coded control points, $\mathbf{x}_c$, with the four user marked coded control points, $\mathbf{x}'_c$, in the image according to

$$\mathbf{H}\mathbf{x}_c = \mathbf{x}'_c \tag{44}$$

where the four virtually coded control points have coordinates, $(2,2)$, $(2,9)$, $(9,2)$ and $(9,9)$. Using the homography the other virtual control points, with coordinates $(1,1),(1,3),...,(10,10)$, can be transformed to their corresponding image points according to

$$\mathbf{H}\mathbf{x}_i = \mathbf{x}'_i \tag{45}$$

see figure 15.

3.2 Region adjustment

Given the approximate centers of the control points, regions containing one control point are extracted from the image, see figure 16. Due to measurement errors, lens distortion etc, the control points may not be centered in the regions, see top left corner of figure 16. Therefor the center of gravity is calculated in

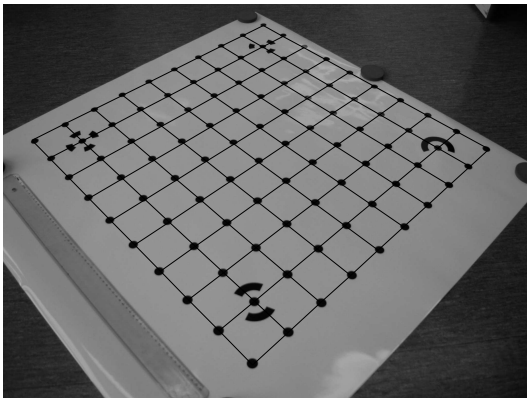


Figure 15: A grid created using an homography placed on the image

each region and this point is used as a more correct position of the control point. Using these points the regions are then updated.

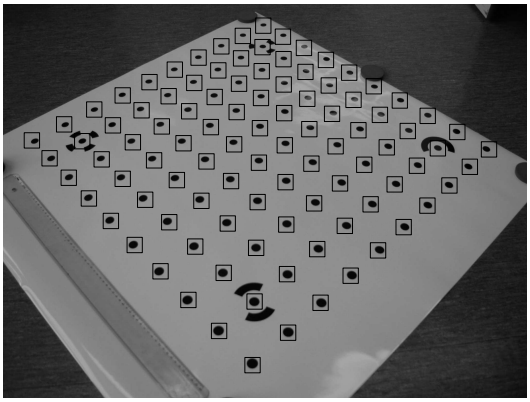


Figure 16: Illustration of how the control points are extracted from the images. Note that the control point in the top left region is not centered.

Since some of the control points are coded, part of the code marks may be left in the extracted regions. Since these markings are not intended and only disrupts the calculations, they are removed.

3.3 Ellipse center calculation using the Centroid method

The centroid is the point where the function is balanced with equal weight on all sides.

$$x_c = \frac{\sum x_i w_i}{\sum w_i} \tag{46}$$

$$y_c = \frac{\sum y_i w_i}{\sum w_i} \tag{47}$$

where x_i,y_i are the coordinates and w_i is the intensity value of the points.

The main advantage of using the centroid is that its easily calculated. But using the centroid in this application presents problems, that is deciding what part of the image is to be included in the calculation. The only part that should be included is the ellipse part of the image. In order to get a good result a way to extract the ellipse part of the image is needed. This is usually done by using a *cut-off value* and keeping all points with value above or below it [6]. The effects of using different *cut-off values* is illustrated in figure 17.

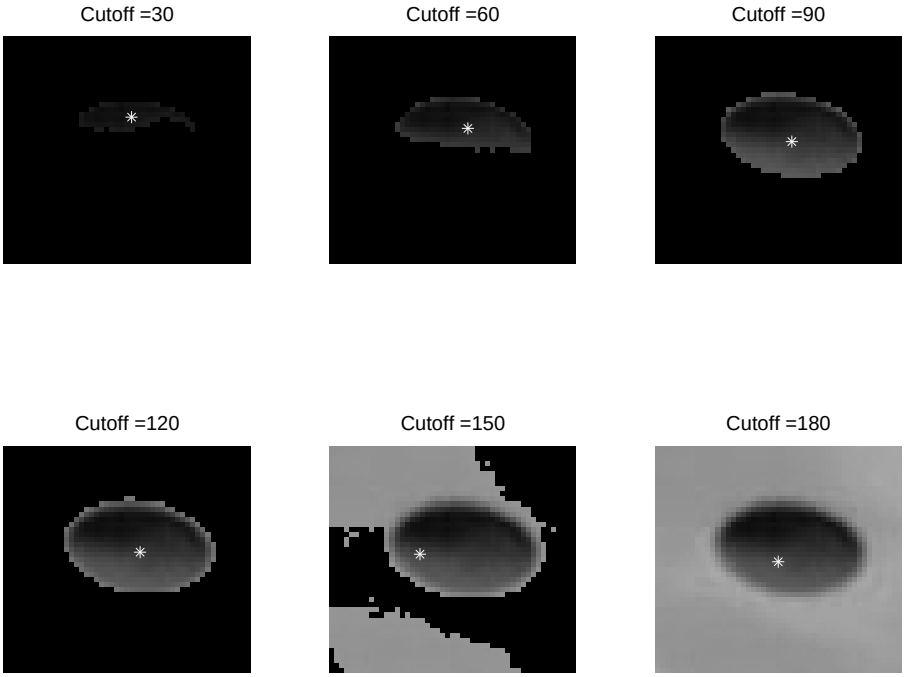
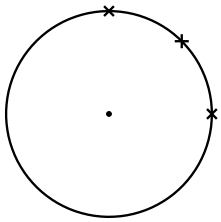
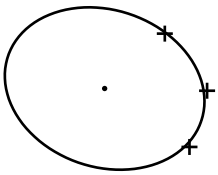


Figure 17: Effects of using different cut-off values when calculating the centroid

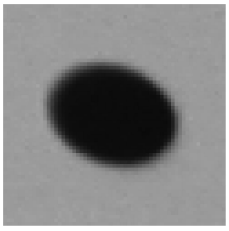
As can be seen in the figure different *cut-off values* give dramatically different results. Of the six examples above only cut-off values 90 and 120 should be considered. To counter this problem of finding the right cut-off value another way to isolate the ellipse part of the image has been implemented. It is done by using the homography created earlier. Beside the circle centers, three more points on the fringe of the circles are mapped to the image. Using these four points an estimation of the ellipses can be calculated. The ellipse estimations is then used as filters on the control point images to blacken out the part that is not the control point. See figure 18 for illustration. The centroid is then calculated on the filtered images.



(a) Circle



(b) Ellipse resulting from applying the homography on the circle



(c) Image of a control point

(d) Control point after the ellipse is used as mask

Figure 18: Illustration of how the ellipse is isolated from the image of the control point

3.4 Ellipse center calculation using a Least Squares method

The second method used is a Least Squares method. Using the theory discussed in section 2.5, a model mimicking the image of the ellipse is used. Two model functions where considered, first a sigmoid based function [2](see figure 19)

$$f_e(x,y) = \frac{k}{1+e^{s(d-1)}} \tag{48}$$

and second an arctangent based function

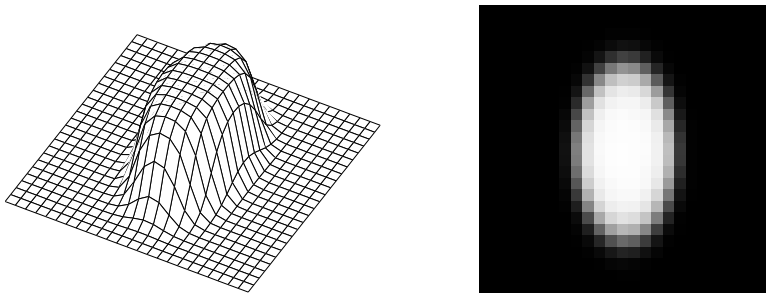
$$f_e(x,y) = k\left(\frac{\pi}{2} - \arctan\left(e^{s(d-1)}\right)\right) \tag{49}$$

In both functions $d = A(x-x_c)^2 + B(y-y_c)^2 + 2C(x-x_c)(y-y_c)$ which represents the ellipse. To model the background of the image a planar function is used.

$$f_p(x,y) = c_0 + c_1 x + c_2 y \tag{50}$$

The full image models is given by

$$f_m(x, y) = f_e + f_p \tag{51}$$



(a) Wire frame (b) Resulting image

Figure 19: The sigmoid model

Using a nonlinear optimization algorithm, the image model functions parameters are optimized and (x_c, y_c) is used as the control point center.

3.5 Circle center calculation using ellipse tangents

As described in section 2.4.1 the ellipse center is not the same as the projected circle center. However after the least squares fitting has been done on all the control points, ellipse parameters are available for each of them, see figure 20.

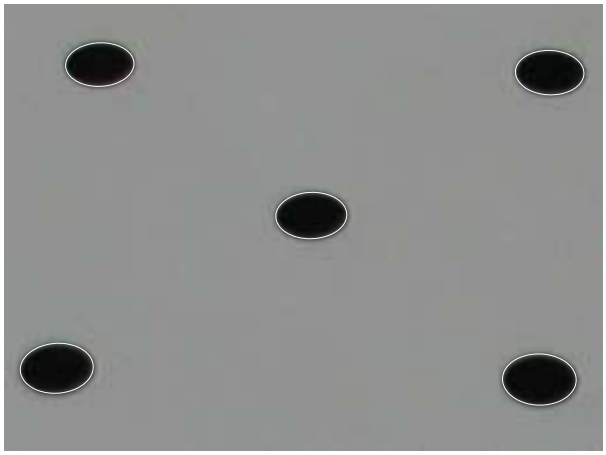


Figure 20: Ellipses resulting from least square fitting of control points

An approximation of the projected circle center is calculated from each neighboring ellipse using the algorithm in section 2.4.1. The number of neighbors vary between two and four depending on the grid position of the control points.

3.6 Identification of outliers using circle center calculation

Some of the control points will be difficult to fit the model to, due to reflections in the image and their ellipse data will therefor be misleading. An example is shown in figure 21. This will create distorted projected circle centers in their neighbors.

Figure 21: Ellipses resulting from least square fitting of distorted control points

Since each control points contribute a circle center in each of its neighbors, all of these circle centers will be bad approximations if the ellipse fitting is bad. So by comparing each of these circle centers from the poorly fitted control point to the other circle centers in the neighbors, it can be identified and removed. For each control point i a wight w_i is calculated

$$w_i = \sum_{j=1}^{n} \parallel \mathbf{c}_{ij} - \overline{\mathbf{c}}_j \parallel \tag{52}$$

where n is the number of neighbors, $\mathbf{c}_{ij}$ is the circle center approximation of control point j calculated using control point i and $\overline{\mathbf{c}}_j$ is the average value of the other circle centers in control point j.

The control points with largest weights are classified as outliers.

3.7 Calibration using centers

After a number of images has been processed, resulting in a large number of 3D-point to 2D-point correspondences, the camera is calibrated using the least squares method.

The calibration method is done using the following camera model

$$\mathbf{P} = \mathbf{KR}[\mathbf{I}| - \mathbf{C}] \tag{53}$$

where

$$\mathbf{K} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$

And compensates for lens distortion by using two parameters for describing radial and two for describing tangential distortion.

4 Investigation

The following methods for control point center detection where compared.

- The centroid method
- The least squares method
- The least squares method compensated for circle center displacement

They where compared with and without outlier detection. In addition they where compared to data collected from PhotoModeler.

The following internal camera parameters where recorder.

f	–	the focal length
pp	–	the principal point
K_1	–	the first radial distortion parameter
K_2	–	the second radial distortion parameter
P_1	–	the first tangential distortion parameter
P_2	–	the second tangential distortion parameter

The following external parameters where recorder.

C	–	the camera position
R	–	the camera rotation

5 Results

An illustration showing the resulting camera positions and rotations relative the grid is shown in figure 22.

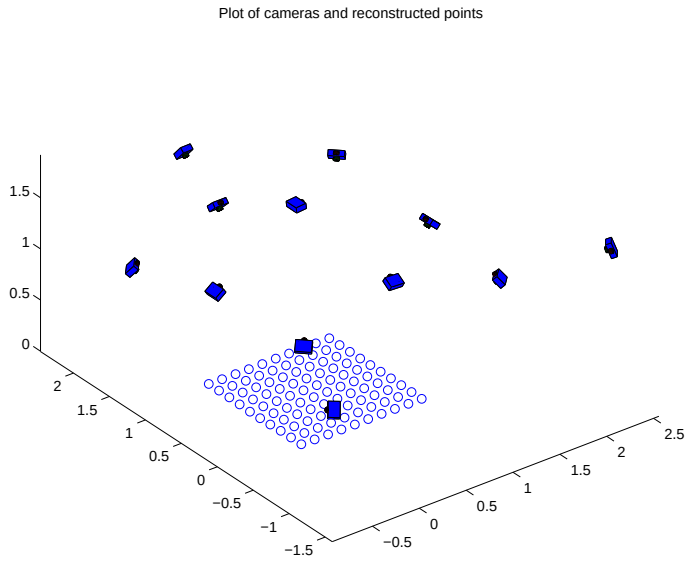


Figure 22: The position of the camera for each image relative to the grid of control points

Tests have been run using recorded values from PhotoModeler, centroid calculated, least squares calculated and circle center compensated centers. The data collected from PhotoModeler is limited to specific points so in order to compare it to the methods implemented, tests have been run using the control points they have in common. The resulting data of this comparison can be seen in table 23. Since there is no correct result, an error measurement is needed. The natural choice is the norm of the residual, $\|x - PX\|$. A table comparing the residual errors can be seen in figure 24.

	PhotoModeler using 845 points	Centroid using 845 points	Least squares using 845 points	Compensated LS
f	7.366	7.370	7.361	7.361
p_x	3.557	3.557	3.561	3.561
p_y	-2.587	-2.584	-2.590	-2.591
K_1	-5.08×10^{-3}	-5.39×10^{-3}	-5.02×10^{-3}	5.02×10^{-3}
K_2	9.68×10^{-5}	1.21×10^{-4}	9.43×10^{-5}	9.40×10^{-5}
P_1	4.87×10^{-5}	4.86×10^{-5}	5.40×10^{-5}	5.33×10^{-5}
P_2	4.95×10^{-5}	8.45×10^{-5}	4.83×10^{-5}	4.64×10^{-5}

Figure 23: Internal camera parameter values after calibration using 845 common control points between PhotoModeler and the implemented methods

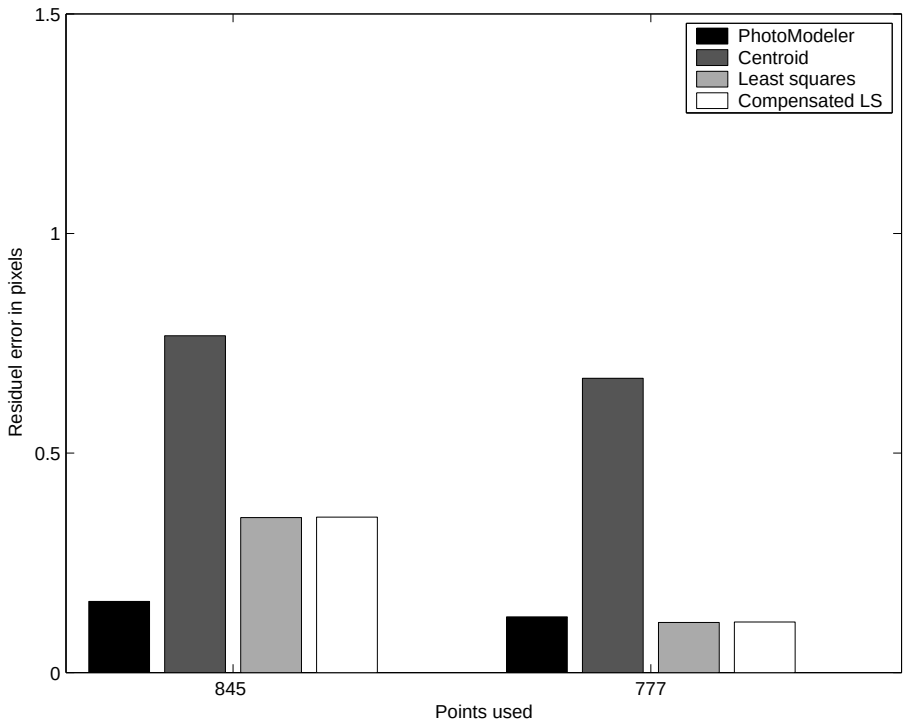


Figure 24: A table comparing the residual error of each method measured in pixels. The two different sensitivity settings, resulting in different number of points used, compose the x-axis.

The three implemented methods have been compared to each other using three different sensitivity settings for identification of outliers. See table 25 for data collected from one of these test. A table comparing the residual errors is shown in figure 26. The results from using the PhotoModeler generated control point centers have also been included for comparison, even though it has been test run with a different set of control points.

	PhotoModeler using 961 points	Least squares using 1039 points	Compensated LS using 1039 points	
f	7.374	7.365	7.358	7.358
p_x	3.557	3.550	3.557	3.557
p_y	-2.579	-2.580	-2.591	-2.592
K_1	-5.13×10^{-3}	-5.30×10^{-3}	-4.99×10^{-3}	-4.98×10^{-3}
K_2	9.92×10^{-5}	1.16×10^{-4}	9.32×10^{-5}	9.2×10^{-5}
P_1	4.75×10^{-5}	1.67×10^{-5}	4.05×10^{-5}	4.02×10^{-5}
P_2	7.23×10^{-5}	1.23×10^{-4}	6.35×10^{-5}	6.06×10^{-5}

Figure 25: Internal camera parameter values after calibration using 1039 control points for the implemented methods and 961 control points for PhotoModeler

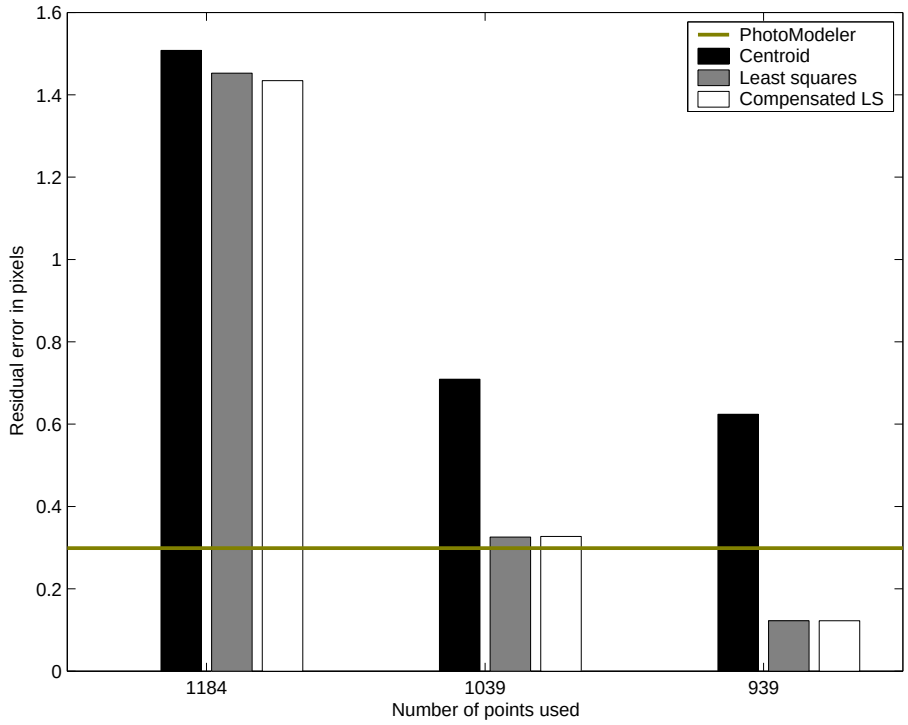
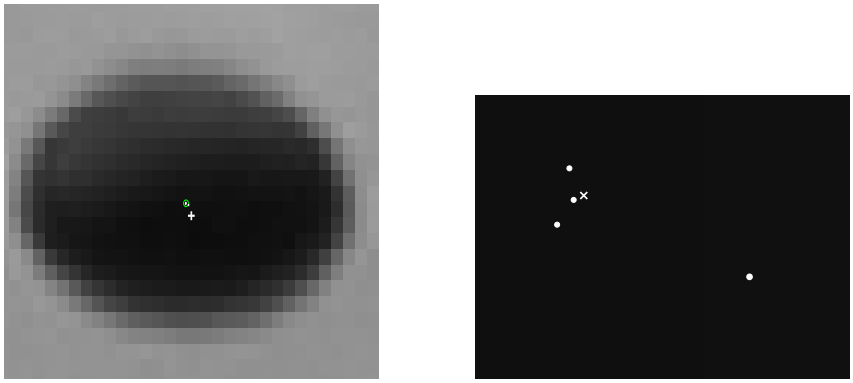


Figure 26: A table comparing the residual error of each method measured in pixels. The three different sensitivity settings resulting in different number of points used compose the x-axis. The first setting, 1185 points, is the result of no outlier removal. PhotoModeler has been test run using 961 control points.

As seen in figure 27 the circle center compensated points proved not to be so different from the ellipse centers. In many cases the distance between the compensated points where larger then the distance to the ellipse center. Using a mean value of these point would produce center approximations that are no more precise then the ellipse centers. One of the compensated circle centers is distant from the others. This is meant to illustrate the fact that the use of the compensated points to determine outliers worked well.



(a) Control point (b) Zoomed 64 times

Figure 27: Image of a control point with the different centers marked. The centroid center is marked with a cross, the ellipse center with a circle and the circle center compensated points with dots.

6 Conclusion

Comparing the new method to the data collected from PhotoModeler is somewhat difficult. Even though tests have been run using the same set of control points the sets are not optimal for either method. The methods most likely differ in which control points they measure accurately and therefor the sets of common control points are not the best for either method. As seen in figure 24 the implemented methods give better results when fewer common control points are used. This is because a stricter setting has been used for outlier removal on the implemented methods resulting in fewer common control points.

In figure 26 the results from the unaltered PhotoModeler data is compared to results from using the implemented methods with various outlier removal settings. Even though PhotoModeler and the implemented methods do not use the same set of control points, the number of control points used can be used for comparison instead. As seen the least squares method give similar results as PhotoModeler using 1039 points and better results when using 939 points. As PhotoModeler use 961 points this could indicate that the least squares method measure the control points more accurately then PhotoModeler.

In comparison to the centroid method implemented, the new method shows a substantial improvement in all the tests. If this due to shortcomings in the implementation of the centroid method or if its because that the new methods is more accurate is hard to tell.

Compensating for ellipse center not being the projected circle center did not improve the results. This is mainly because there is less distortion then expected and in light of this the least squares method is not as precise as needed. Non of the tests show better results when using the compensated points. Using the compensation for identification of outliers works very well but since its not worth using for more exact calibration, other less calculation expensive methods could be used. For instance outliers could be identified by looking at the control points residuals after each fitting.

Overall the least squares method is a lot more expensive in turns of calculation complexity then the centroid method. But since a camera does not have to be calibrated so many times this is not such a big problem.

7 Acknowledgments

I would like to thank my supervisor Niclas Börlin for great guidance, help and patience with this thesis. I would also like to thank family and friends for support and for listening to endless monologues on ellipses.

References

[1] K. B. Atkinson, editor. *Close Range Photogrammetry and Machine Vision.* Whittles Publishing, 2001.

[2] Niclas Börlin. *Model-Based Measurements in Digital Radiographs.* PhD thesis, Department of Computing Science, Umeå Universitet, june 2000.

[3] Vincent Fremont and Ryad Chellali. Direct camera calibration using two concentric circles from a single view. In *ICAT '02: International Conference on Artificial Reality and Telexistence, Tokyo*, 2002.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in computer vision.* Cambridge University Press, 2002.

[5] Michael T. Heath. *Scientific Computing - An introductory survey.* McGraw-Hill, 1997.

[6] Janne Heikkila and Olli Silven. Calibration procedure for short focal length off-the-shelf ccd cameras. In *13th International Conference on Pattern Recognition, Vienna, Austria, p. 166-170*, 1996.

[7] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), San Juan, Puerto Rico, p. 1106-1112.* IEEE Computer Society, 1997.

[8] A. Howard and C. Rorres. *Elementary Linear Algebra - Applications version.* John Wiley & Sons, 7 edition, 1994.