We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

18

Camera Modelling and Calibration - with Applications

Anders Ryberg, Anna-Karin Christiansson,
Bengt Lennartson and Kenneth Eriksson
University West
Sweden

1. Introduction

Methods for measuring/calculating positions and poses using vision and 2D images are presented in this chapter. In such calculations a calibrated camera model is needed, and a newly developed generic camera model (GCM) is proposed and presented together with calibration routines. A camera model is a mapping of a 3D object space to a 2D image space and/or vice versa. This new GCM is aimed to be more accurate, computationally efficient, flexible and general compared to conventional camera models. See Fig. 1 for an application. The new camera model was developed because measurements showed that a conventional camera model (CCM) did not have a high enough accuracy for some camera types and robot positioning applications.

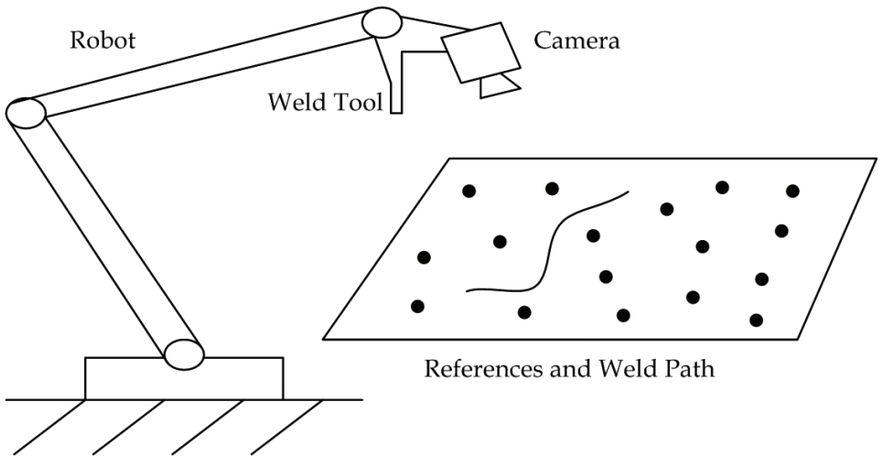

Fig. 1. A vision system can measure the pose of a robot if the camera can see references. It can also determine the geometry of a curve to be welded by the robot. In that case it needs to first see the path from at least two directions and use stereo vision.

The calibration calculates intrinsic and extrinsic camera parameters as well as positions of references from several images of these, using optimization methods. The extrinsic camera parameters are the 6D pose of the camera. The intrinsic camera parameters determine how the 2D image is formed in the camera given relative positions of the camera and the environment.

Source: Computer Vision, Book edited by: Xiong Zhihui,
ISBN 978-953-7619-21-3, pp. 538, November 2008, I-Tech, Vienna, Austria

Methods for estimating the camera parameters and the reference positions are presented. These calculations are based on the position, size and shape of references in the images. Estimates of these parameters are needed as initial values in the calibration to assure convergence. A method of calculating the "centre point" of a reference in the image is developed for increased accuracy, since the centre of gravity of the reference in the image generally does not correspond to the centre of the reference. The GCM allows for variable focus and zoom. This fact and that it can be used for wide angle fisheye cameras (which can not be modelled by the CCM) as well as low distortion cameras makes the GCM very versatile.

Together with the GCM and the calibration, ambiguities or nontrivial null spaces are also discussed. Nontrivial null spaces occur when the same image can be obtained with different sets of camera parameters and camera poses. This can happen with the GCM, as well as the CMM to some extent, and is therefore important to consider in the calibration. Methods to handle these null spaces are described.

Image processing techniques are not explicitly described, except for methods of finding centre points of the references in the image. It is assumed that the image coordinates needed are found with some method.

Different kinds of distortions can be captured by a camera model. In a camera with no distortion, a collinear camera, a straight line in object space is mapped to a straight line in image space. A model of a collinear camera is called a pinhole camera model (PCM). Both the CCM and the GCM can model ordinary radial distortion, but in different ways, while other types of distortions are modelled by some CCMs and the GCM, e.g. variations in entrance pupil (as in (Gennery 2006)) and decentring distortion (more in (Heikkila 2000) and (Kannala & Brandt 2006)). The GCM has an efficient way of including varying entrance pupil in the model, and two methods for compensating for decentring distortion. The CCM, described by (Brown 1971), (Tsai 1987), (Heikkila 2000), (Motta & McMaster 2002), and the GCM, have been used in a laboratory experiment, where the accuracy of the vision systems were compared to a coordinate measuring machine (CMM). When only radial distortion is accounted for the GCM is better in accuracy. Then the other types of distortion can be added for further improvement of accuracy. Furthermore, the CCM is poor in handling cameras with a wide angle of view, such as fisheye cameras, which is why there are several other models developed, specialised for these (Basu & Licardie 1995), (Brauer-Burchardt & Voss 2001), (Bakstein & Pajdla 2002), (Shah & Aggarwal 1996). The GCM can be used for both fisheye and ordinary cameras, and at the same time it includes the undistorted PCM as the simplest meaningful special case. Therefore there is no longer a need to use different models for these different kinds of cameras.

Calibration methods using references which are a priori known can be found in (Brown 1971), (Tsai 1987) and (Heikkila 2000). If the positions of the references are not known the calibration procedure is called self-calibration. Both self-calibration and calibration with known references are presented in this chapter. Self calibration procedures can also be found in (Fraser 1997) and (Dornaika & Chung 2001). Other methods for self-calibration use modulus constraint (Pollefeys & Van Gool 1999) and Kruppa's equations (Olivier et al. 1992), the last two use only the undistorted PCM. Another calibration method that uses external measurements of the calibration camera poses is (Wei et al. 1998). Such a method is called active, while passive methods only use the information in the images for the calibration. Both passive and active methods are discussed below, and a method of improving the calibration by measuring camera positions after a passive calibration, is presented.

A calibrated camera can then be used for calculating a camera pose from an image, by using references with known positions. If a point or curve is seen from at least two directions its position can be calculated, using stereovision methods. A new general method for calculating positions from stereo vision is presented.
There are other models that can be considered generic, as in (Kannala & Brandt 2006) and (Gennery 2006). An advantage with the GCM compared to (Kannala & Brandt 2006) is that they do not include the undistorted PCM as a special case, and also do not have entrance pupil variations included. Advantages compared to (Gennery 2006) are that GCM is more simple and efficient, both to implement and to run. (Gennery 2006) needs several iterations to do one camera model mapping, while the GCM can do it in a single strike. Fast calculations are important for both calibration and the applications; in the calibration since it involves many camera model calculations, and in the applications if the result is to be used on-line.
To summarize the novelties of this contribution are as follows:

- Introduction of a generic camera model (GCM) and its different kinds of distortion compensations, as well as its relation to other camera models. A main benefit of the GCM is that it includes wide angle of view (fisheye) cameras as well as ordinary cameras within the same unified model structure.
- Methods for including variable zoom and focus in the model as well as procedures for how to include it in the calibration.
- Discussions on nontrivial null spaces and how to handle them.
- Algorithms for initial estimates of intrinsic and extrinsic camera parameters as well as reference positions.
- Methods for calculating image centre points of references.
- A new stereo vision calculation method.
- Experimental investigation where the accuracy of different camera model configurations are analysed.

The chapter is organised as follows; Section 2 presents camera models in more detail, both the CCM and especially the GCM. Section 3 describes calibration while Section 4 introduces vision pose calculations and stereo vision. Section 5 presents an accuracy investigation, and conclusions and future work are given in Section 6.

2. Camera models

A mathematical camera model consists of algorithms for conversion between the position of points in a 3D object world and their appearance as points in a 2D image plane. If the intrinsic and extrinsic camera parameters are known and an observed 3D object point position is known, the camera model can consequently be used to calculate where the object point ends up in the image. It can also be used the other way around; if an image point and the camera parameters are known the camera model can calculate all possible object points the image point could originate from.
Both the CCM and GCM are described here. First an image coordinate system is introduced, where the origin is in the intersection of the image plane with the optical axis, called the principal point. The scaling is the same in both image coordinate directions. A conversion to the detector pixel image coordinate plane is made in Section 2.3. First only radial distortion is considered and decentring distortion can be added afterwards. Vectors with a superscript

^{w} are coordinates in a world coordinate system, superscript ^{i} in a 2D image coordinate system, and superscript ^{c} in a 3D camera coordinate system. Indices $_1$, $_2$ and $_3$ denote *x*- *y*- and *z*- components of a vector respectively.

2.1 Conventional Camera Model (CCM)

In the simplest kind of camera model, the pinhole camera model (PCM), a point in 3D space is projected to the image plane through a straight line passing a point P inside the lens system on the optical axis, see image point x_p^i in Fig 2. This model is collinear and takes no distortion into consideration. To calculate an image coordinate point, x_p^i, corresponding to an observed object point, x_o^w, using this model, first the points coordinates in a 3D camera coordinate system, x_o^c, should be calculated, by rotating and translating the coordinates according to a camera pose. The pose is the position and orientation of the camera

$$x_o^c = Rx_o^w + T \tag{1}$$

R is a rotation matrix and T is a translation vector of the conversion defining the pose of the camera, that is rotation and translation between a world and a camera coordinate system. The camera coordinate system has its *x*- and *y*- axes parallel to the image coordinate axes, and its *z*-axis along the optical axis. Its origin is in a point *P*, in the centre of the lens system, so this point is defined as the camera position for PCM and CCM, see Fig. 2. Note that the definition of the camera position differs for the GCM in the next section. For the GCM the origin of the camera coordinate system is the principal point, that is the intersection between the optical axis and the image plane. The image coordinates for a PCM are now

$$x_{p1}^i = -f\, x_{o1}^c / x_{o3}^c \tag{2a}$$

$$x_{p2}^i = -f\, x_{o2}^c / x_{o3}^c \tag{2b}$$

where *f* (related to the focal length) is the distance from the detector to the pinhole point *P*. Sometimes a geometrical construction where the image is formed in front of the detector is used, and then the minus signs in (2) change to plus signs.

One way of modelling distortion is to use a conversion between an image point and where it would end up in a PCM, i.e. a conversion between a distorted and a non distorted image. A CCM models radial distortion by taking a PCM point x_p^i, and moving it in the radial direction to the point x_r^i where it would end up in a radially distorted image, see Fig 2. A polynomial in $r_p = \sqrt{x_{p1}^{i\,2} + x_{p2}^{i\,2}}$, the distance in the image from the image point to the principal point, is used to adjust the position of the image point, according to

$$x_r^i = x_p^i f_p(r_p) \tag{3}$$

$$f_p(r_p) = (1 + k_{p1} r_p + k_{p2} r_p^{\,2} + \ldots) \tag{4}$$

This polynomial adjusts the image point radially from or to the principal point. The constants k_{pi} are intrinsic camera parameters and the degree of the polynomial $f_p(r_p)$ can be adjusted according to the camera used and the accuracy needed. Other functions than

polynomials can be used in (4), but a polynomial was chosen for simplicity. Sometimes only even powers of r_p are used. Note that from (3) it follows that

$$r_r(r_p) = r_p f_p(r_p) \tag{5}$$

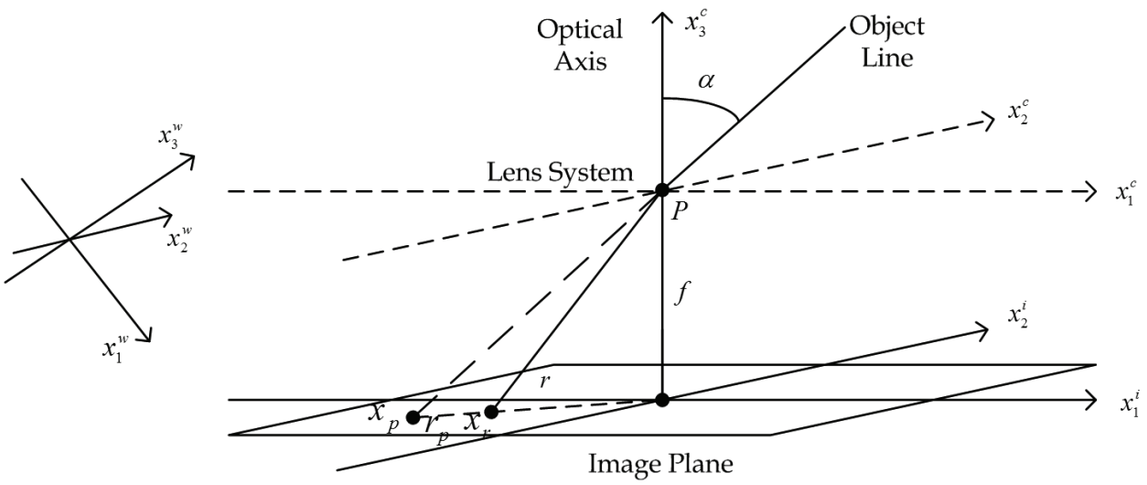

Fig. 2. Illustration of the CCM with the world, camera and image coordinate systems. Two points in the image are defined, one undistorted, x_p, and one radially distorted, x_r.

In (5) r is the distance to the principal point in the radially distorted image, $r_r = \sqrt{x_{r1}^{i\,2} + x_{r2}^{i\,2}}$. If every image point is moved radially according to (5) this is the same as (3). The points are moved radially if the ratio between x_1^i and x_2^i is the same before and after the transformation. The transformation (3) or (5) can be performed in the opposite direction, so that

$$x_p^i = x_r^i f_r(r_r) \tag{6}$$

where $f_r(r_r) = (1 + k_{r1} r_r + k_{r2} r_r^{\,2} + ...)$. Also here, usually only even powers of r_r are used. These methods are described in (Brown 1971), (Tsai 1987), (Swaminathan & Nayar 1999), (Heikkila 2000), (Motta & McMaster 2002) and (Nowakowski & Skarbek 2007). If there is a need to calculate from the 2D image to 3D object space an equation for a straight line from the undistorted image coordinate converted to the 3D coordinate system through the pinhole point, P, is made. Then an image conversion from the distorted to non distorted image is needed. So whether (5) or (6) should be used depends mostly on the direction of the camera model transformation. When transforming to the image only the function value needs to be computed in (5), while in the other direction a polynomial equation has to be solved. With (6) it is on the contrary easier to convert from distorted to non-distorted image. The inclination angle α in Fig 2 is between the optical axis oaches 90 degrees, therefore large angles can not be modelled with the CCM. and the object line. One problem with the CCM is that r_p tends to infinity when α approaches 90 degrees, therefore large angles can not be modelled with the CCM.

2.2 The new Generic Camera Model (GCM)

The GCM handles radial distortion in a different way than the CCM, and can also include variation of the entrance pupil position and decentring distortion. Entrance pupil can be viewed as another kind of radial distortion than discussed in the previous section. To explain entrance pupil, think of an image point. That corresponds to a line in the 3D object space, corresponding to all points the image point can have originated from, the object line. The line crosses the optical axis, but it can cross it on different positions depending on the inclination angle. The crossing point is called the entrance pupil and is denoted x_{fo}, see Fig 3. The GCM and the different distortions are divided into steps where each step accounts for one type of distortion. All of these steps can be performed in two directions either adding or removing the distortion type. Adding distortion is done when converting from object space to image space. In converting from image to object space the distortions need to be removed. If the different distortions are compensated for in the right order they can be separated into different steps, except for radial distortion and entrance pupil, which are connected, so they have to be handled simultaneously. First a conversion between a 3D object world and an image with radial distortion and varying entrance pupil, x_r^i is introduced. Then a conversion between x_r^i, the radially distorted image point, to a decentring distorted point, x_d^i, is performed. Finally, a conversion between x_d^i and x_c^i, where, x_c^i is the measured chip pixel coordinates, is presented. The undistorted PCM coordinates x_p^i can also be computed using the GCM as in (28). In converting to the detector the calculations should be in this order. When converting to object space the calculations are done in reverse, i.e. from x_c^i to x_d^i, then from x_d^i to x_r^i and from x_r^i to the object line. Variable focus and zoom in the model is presented in the calibration section.

Radial Distortion and Entrance Pupil

Again we start by using a 2D image coordinate system where the origin is in the principal point. The 3D camera coordinate system has its origin in the same point, that is in x_{ca} in Fig 3, and not in P (= x_{fo}), as for the CCM. So x_{ca} is the position of the camera in the GCM. We start by converting from image to object space. In calculating with radial distortion and entrance pupil with the GCM first two points are defined on the optical axis, called x_{fi} and x_{fo}, see Fig 3, where x_{fo} already has been discussed. Both of these points can slide along the optical axis depending on the inclination angle, α, or depending on *r* which is a measure of α. The object line is parallel to a line from an image point, x_r, to x_{fi}, and it goes through x_{fo}. This geometrical construction defines the radial distortion and the entrance pupil of the GCM, see Fig 3.

The distance from the detector to the points x_{fi} and x_{fo} are called $f_{inner}(r)$ and $f_{outer}(r)$. *r* is the distance from the image point to the principal point as before. The dependence of *r* in f_{inner} and f_{outer} can be set to polynomials,

$$f_{inner}(r) = d_0 + d_1 r + d_2 r^2 + \ldots \tag{7}$$

$$f_{outer}(r) = e_0 + e_1 r + e_2 r^2 + \ldots \tag{8}$$

where d_i and e_i are intrinsic camera parameters. Here d_0 is the same as f in the CCM. The degrees of the polynomials can be chosen to get a good enough accuracy and a simple enough model. It might seem more natural to have f_{inner} and f_{outer} as functions of α instead of *r*, but since α is not directly known, that would give a more complicated and slower calculation. The reason that it is possible to use *r* instead of α is that for a given camera there is a one to one relationship between them and that we have the freedom to design an appropriate function dependence (by choosing polynomial coefficient values) between f_{inner} and f_{outer} and its variable. Compare with (Gennery 2006), which gives more complicated formulas for modelling the same phenomena.

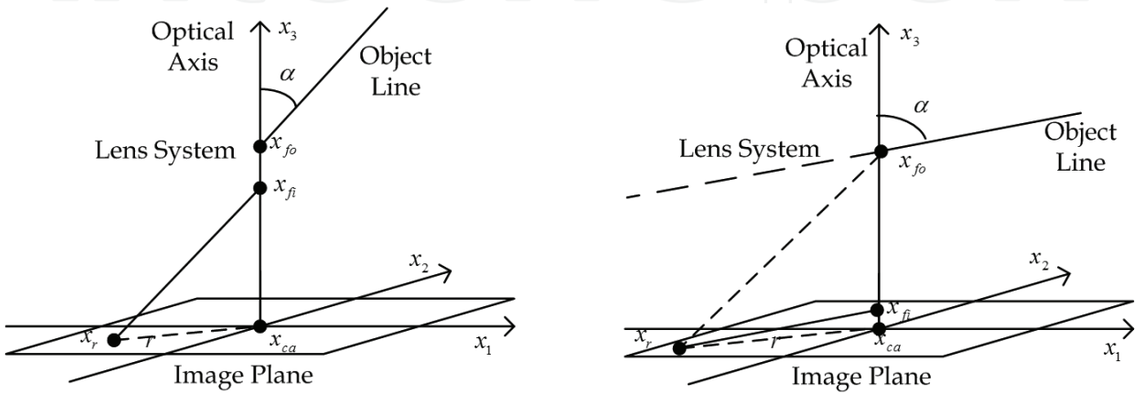

Fig. 3. Geometrical construction of the GCM, with different inclination angles. The points x_{fo} and x_{fi} can slide along the optical axis depending on *r*, the distance to the centre principal point. For large angles, α, the undistorted image point of the CCM would be far away from the centre of the image, causing problems for the CCM. The GCM solves that by having x_{fi} close to the detector (or even below it).

Let the unit vectors e_x, e_y and e_z span a 3D camera coordinate system. The *x*- and *y*- axes in this 3D camera coordinate system are the image *x*- and *y*- axes, and the *z*- axis is pointing along the optical axis. If the camera is rotated by angles θ, φ and γ compared to a world coordinate system, the unit vectors in the world coordinate system can be expressed as

$$e_z^w = \begin{bmatrix} \cos\theta\cos\varphi \\ \cos\theta\sin\varphi \\ \sin\theta \end{bmatrix};\quad e_x^w = \begin{bmatrix} \cos\gamma\sin\varphi + \sin\gamma\sin\theta\cos\varphi \\ \cos\gamma\cos\varphi + \sin\gamma\sin\theta\sin\varphi \\ -\sin\gamma\cos\theta \end{bmatrix} \tag{9ab}$$

$$e_y^w = e_z^w \times e_x^w = \begin{bmatrix} \sin\gamma\sin\varphi + \cos\gamma\sin\theta\cos\varphi \\ -\sin\varphi\cos\varphi + \cos\gamma\sin\theta\sin\varphi \\ -\cos\gamma\cos\theta \end{bmatrix} \tag{9c}$$

Relations between the points in Fig 3 and the unit vectors and the polynomials (7,8) in a world coordinate system are

$$x_{fo}^w = x_{ca}^w + f_{outer}(r)e_z^w \tag{10}$$

$$x_{fi}^w = x_{ca}^w + f_{inner}(r)e_z^w \tag{11}$$

$$x_r^w = x_{ca}^w + x_{r1}^i e_x^w + x_{r2}^i e_y^w \tag{12}$$

The object line can now be expressed as

$$x_{fo}^w + a(x_{fi}^w - x_r^w) \tag{13}$$

where the parameter, *a*, can be varied to move along the object line. Using (10,11,12) the object line (13) can also be written

$$x_{ca}^w + f_{outer}(r)e_z^w + a(f_{inner}(r)e_z^w - x_{r1}^i e_x^w - x_{r2}^i e_y^w) \tag{14}$$

The terms inside the parenthesis after *a* defining the direction of the line should be measured in the same unit, where an image unit like pixels is convenient. The two first terms should be in the length unit of a world coordinate system, not necessarily the same as inside the parenthesis. Equations (7)-(14) represent the conversion from the 2D image to the 3D object world. Using this method it is also possible to go in the other direction, i.e. from the object space to the image space. Assume we have a point in the object space, x_o^w, with a certain position relative to the camera. First calculate the point's position in the 3D camera coordinate system, x_o^c, using (1). Then the following equation for *r* can be used, derived using similarity of triangles:

$$\frac{f_{inner}(r)}{r} = \frac{x_{o3}^c - f_{outer}(r)}{\sqrt{x_{o1}^{c\,2} + x_{o2}^{c\,2}}} \tag{15}$$

This is an equation that can be solved for *r* so the distance to the centre of the image is known. Then use the fact that if there is no decentring distortion the ratio between x_{r1}^i and x_{r2}^i in the image is the same as between x_{o1}^c and x_{o2}^c, but they have opposite signs (or the same signs if it is projected in front of the lens). This gives the following formulas for x_r^i based on the solution of (15) and the vector x_o^c

$$x_{r1}^i = -r\frac{x_{o1}^c}{\sqrt{x_{o1}^{c\,2} + x_{o2}^{c\,2}}} \tag{16a}$$

$$x_{r2}^i = \begin{cases} x_{r1}^i \dfrac{x_{o2}^c}{x_{o1}^c}, & x_{o1}^c \neq 0 \\ -\operatorname{sign}(x_{o2}^c)r, & x_{o1}^c = 0 \end{cases} \tag{16b}$$

Since (15) can be transformed into a polynomial equation if $f_{inner}(r)$ and $f_{outer}(r)$ are polynomials it can have several solutions. If more than one is real a test has to be made to obtain the correct solution. *r* should have appropriate values in relation to the size of the detector. It can also be tested if the corresponding values of $f_{inner}(r)$ and $f_{outer}(r)$ are reasonable. The degree of the equation (15) is

$$\text{degree(eq 15)}=\max((\text{degree } f_{inner}(r)\,);\ (\text{degree } f_{outer}(r)\,)+1) \tag{17}$$

Therefore, if conversion shall be done to the image, the degree of $f_{outer}(r)$ should usually be at least one lower than $f_{inner}(r)$ so that it does not contribute to give a higher degree of the polynomial equation. This is not a problem since it is more important to have an accurate $f_{inner}(r)$ than $f_{outer}(r)$. If $f_{outer}(r)$ is constant and a low order polynomial in (15) is wanted, in relation to the number of camera parameters, then a quotient between two polynomials can be used as $f_{inner}(r)$, where the degree of the denominator polynomial is one degree lower than the numerator.

Decentring Distortion

Decentring distortion can be caused by e.g. a leaning detector surface, badly aligned lens system and non constant refraction index in the lenses. These effects are usually larger for cheap cameras. A special method accounting for leaning detector is presented first, and then a more general method will be presented. Leaning detector is compensated for by using formulas defining a conversion between points in a leaning detector and a non-leaning detector. So now we convert between x_r and x_d. The compensation for leaning detector is such that a straight line between a point in the non-leaning and the corresponding point in the leaning detector crosses the optical axis at a distance $f_l(r)$ from non-leaning image plane, Fig 4.

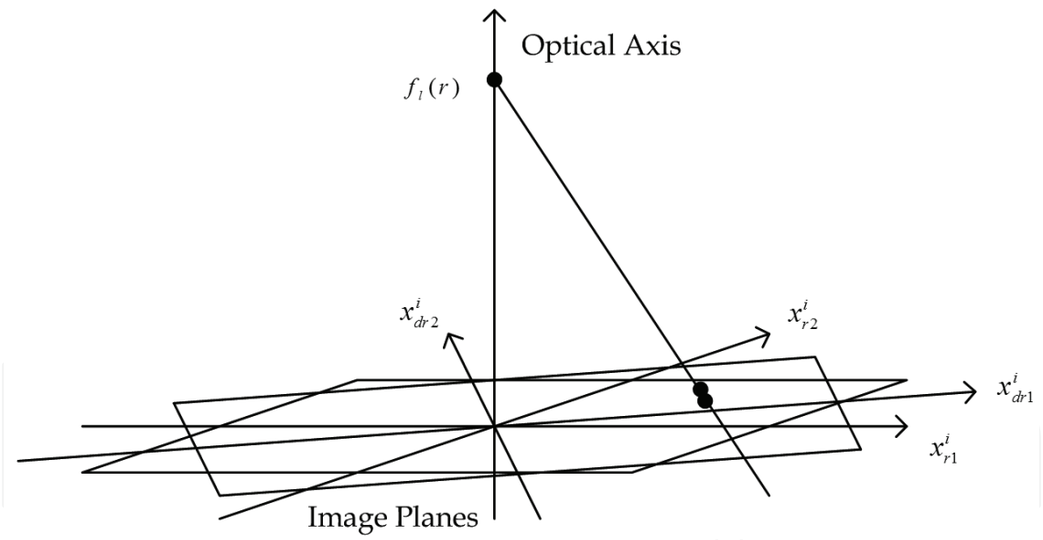

Fig. 4. The leaning image compensation converts between leaning and non-leaning detectors so that a line between the points on the two planes hits the optical axis at a point $f_l(r)$ units from the principal point.

First it is assumed that the image coordinate system is rotated, by an angle β, so that the *x*-axis is pointing in the direction of the steepest descent of the leaning detector, and the centre of the image is still in the principal point. The point x_r^i is called x_{rr}^i in the rotated image coordinate system. If δ is the leaning angle of the detector, the relations between the coordinates in the two planes are obtained from geometric analysis as

$$\frac{\cos(\arctan(x^i_{rr1}/f_l(r)))}{x^i_{dr1}} = \frac{\cos(\delta + \arctan(x^i_{rr1}/f_l(r)))}{x^i_{rr1}} \tag{18a}$$

$$\frac{\cos(\arctan(x^i_{rr2}/f_l(r)))}{x^i_{dr2}} = \frac{\cos(\arctan(x^i_{rr2}/f_l(r) + \arctan(x^i_{rr1}\tan\delta/x^i_{rr2}))}{x^i_{rr2}} \tag{18b}$$

Here $f_l(r)$ is a polynomial defining a point on the optical axis similar to $f_{inner}(r)$ and $f_{outer}(r)$. r is the distance to the principal point in the non-leaning image plane, just like before. x^i_{dr} is the image coordinates in the leaning image plane. This leaning image plane should be rotated back so that the orientation of the image plane coordinate is equivalent to the orientation before the leaning detector compensation. Since the plane is now leaning it is better to rotate back a slightly different value than β. If it is desirable to have the optical axis and images x- axes before and after the transformation in the same plane, a relation between the rotation angles is

$$\beta_\delta = \arctan(\cos\delta \tan\beta) \tag{19}$$

Here β is a rotation in the non leaning plane and β_δ is the corresponding rotation in the leaning plane. Note that this gives a relation between the rotation angles, but the rotations should be in opposite directions.

The result of the back rotation, called x^i_d, is then the coordinate in the leaning detector. With (18) it is easy to obtain a closed expression converting from non-leaning to leaning plane, but the other direction is more difficult, because then two equations have to be solved. One simplification conversing from leaning to non-leaning is obtained by using r in the leaning detector as approximation to r in the formulas. If this does not give an accurate enough result an iteration can be used, so that an r from the solution of the approximated equation is used, and solve the system again, giving a better and better value for r. This can be done any number of times, but it should converge quickly. If $f_l(r)$ is constant these iterations are not needed.

Another way of solving the equations (18) from leaning to non-leaning is to use vector algebra. Construct a line going from an image point in the leaning detector plane to a point on the optical axis defined by $f_l(r)$, and solve a linear equation of where this line crosses a non-leaning image plane. One difficulty here is again that r in the non-leaning plane is not known. Again it can be approximated by r in the leaning plane, and if necessary perform iterations to get a better r. So we have methods for converting in both directions.

Leaning detector compensation may also be useful for modelling inaccurately aligned lenses. Another method, taking decentring distortion into account, can be used if there is a combination of different kinds of decentring distortion, or if it is not known what causes the decentring distortion. The method uses a conversion between an image plane with no decentring distortion and an image with decentring distortion, or the other way around. ϕ is an image angle around the optical axis, and r is as before the distance to the principal point from the image point. e^i_r is a unit vector pointing radially away from the centre of the image to the image point. e^i_t is a unit vector perpendicular to that but still in the image plane. Formulas for converting from non-decentring distortion coordinates x^i_r to an image with decentring distortion x^i_d are then

$$d_r(r,\varphi) = (g_1 r + g_2 r^2 + g_3 r^3)(g_4\cos\varphi + g_5\sin\varphi) + (g_6 r + g_7 r^2)(g_8\cos(2\varphi) + g_9\sin(2\varphi)) \tag{20}$$

$$d_t(r,\varphi) = (h_1 r + h_2 r^2 + h_3 r^3)(h_4 \cos\varphi + h_5 \sin\varphi) + (h_6 r + h_7 r^2)(h_8 \cos(2\varphi) + h_9 \sin(2\varphi)) \quad (21)$$

$$d_{tot}(x_r^i) = d_r(r,\varphi)e_r^i(\varphi) + d_t(r,\varphi)e_t^r(\varphi) \quad (22)$$

$$x_d^i = x_r^i + d_{tot}(x_r^i) \quad (23)$$

Here d_r is a distortion in the radial direction and d_t is a distortion in the tangential direction. This is a bit similar to (Kannala & Brandt 2006), but *r* is used instead of α for the same reason as above, and also odd powers are included. Another difference is that the ϕ and *r* dependences are not separated here, as it is in (Kannala & Brandt 2006). More parameters can easily be added or removed, so it describes a family of decentring distortion compensation methods. If there is a need to go more efficiently in the direction from distorted to not distorted image a more suitable method can be achieved by just changing x_r^i and x_d^i in (20-23), and use unit vectors pointing radially and tangentially in the distorted image, then the values of the constants g_i and h_i in (20) and (21) will change. There are other ways to take care of decentring distortion (Brown 1971), (Swaminathan & Nayar 1999), (Heikkila 2000). These types of decentring distortion can also be combined with the GCM. One advantage of the calculations presented here is that the same behaviour can be achieved independently of the directions of the image coordinate axes.
To summarise, with decentring distortion we have methods that efficiently can go from either decentring distorted image plane to non-decentring distorted or vice versa. If there is a need to go in both directions one of the directions will be a bit slower and more complicated.

2.3 Image plane coordinate conversions

In the equations so far it has been assumed that the image coordinate system has its origin in the principal point. Also the same coordinate axis units are used in the two image directions. In a real camera however, this is usually not the case, but that problem is easily solved by having a conversion between the real camera detector chip coordinate system and the simplified ones used above. This is needed both for the CCM and the GCM. The transformation between the coordinate systems, i.e. from a point x_d^i to a detector chip coordinate, x_c^i, is

$$x_c^i = \begin{bmatrix} m_1 & s \\ 0 & m_2 \end{bmatrix} x_d^i + x_{c0}^i \quad (24)$$

Here m_1 and m_2 adjust the image unit scales in *x*- and *y*- direction of the detector. They are different if the pixel distances are different in the two image directions, the ratio between them is called the aspect ratio. x_{c0}^i is the detector pixel coordinates of the principal point. (24) shifts the origin of the coordinate systems, so that it is in the sensor coordinate system origin. If the detector image coordinate axes are not perpendicular to each other the parameter s is used, otherwise it is zero.

2.4 Comparison between the models

The models can be compared by looking at a simple version of the GCM, with constant entrance pupil and no decentring distortion. In that case the CCM and the GCM both model a camera with ordinary radial distortion. The relation between the angle α and r and f are, for PCM, CCM and the GCM respectively,

$$\text{Pinhole Model, PCM } \tan\alpha = \frac{r}{f} \tag{25}$$

$$\text{Conventional Camera Model, CCM } \tan\alpha = \frac{r_p(r)}{f} \tag{26}$$

$$\text{Generic Camera Model, GCM } \tan\alpha = \frac{r}{f_{inner}(r)} \tag{27}$$

Here it can be seen that if $r/p(r)$ where $p(r)$ is a polynomial in r was used in the CCM instead of $r_p(r)$, the same radial distortion model can be obtained as the GCM with constant entrance pupil. The other way around the GCM can be equivalent to a CCM if $f/p(r)$ is used as f_{inner}. A mix between the models can be constructed if a quotient between polynomials is used as $f_{inner}(r)$ or $r_p(r)$. That will also give polynomial equations for the projection to the image (15) for the GCM, (this is true even if also variation in entrance pupil is used in the model, if f_{outer} is a polynomial or quotient between polynomials).

The fact that $\tan\alpha$ is large or even goes to infinity as α approaches 90 degrees is a problem for the CCM, since that can not be modelled by its polynomial trying to compensate that. It is no problem for the GCM, since if $f_{inner}(r)$ is zero also the right hand side of (27) goes to infinity. If $f_{inner}(r)$ is small or even negative it is no problem for the object line of GCM to be directed in larger angles α.

Sometimes there is a need to know how the image would look without distortion. That can be done even if the camera is calibrated (see Section 3) with the GCM, if f_{outer} is constant. The undistorted image can be calculated from similarity of triangles according to:

$$r_p = \frac{f_0 r}{f_{inner}(r)} \tag{28}$$

Every point should be moved radially from the centre according to (28) to get the corresponding undistorted image. Any value of the constant f_0 gives an undistorted image, but if the image scaling in the centre of the image should be preserved, the first constant in the $f_{inner}(r)$ polynomial, d_0, should be used as f_0. Then the image scale will be the same as for the CCM after radial distortion compensation. If decentring distortion was used first the non decentring distorted image points should be calculated and then use (28) on the resulting image. If f_{outer} is not constant this formula will only be an approximation.

Entrance pupil variations are more important for cameras with a high angle of view. Also it is more important if the distance between the camera and the observed objects can vary. That is since, if the distance is not varying, a camera model with a slightly wrong position of

the entrance pupil can approximate an exact model in a good way, see Fig 7. Fisheye cameras have a large depth of focus. Therefore there are three reasons for using the GCM for fisheye cameras. The large viewing angle and the depth of focus makes entrance pupil important, and the way to handle ordinary radial distortion is suitable for fisheye cameras. Since the simplest meaningful special case of the GCM is the PCM, it is suitable for low distortion cameras as well. Therefore the GCM has the best of both worlds compared to the CCM and models specialised for fisheye lenses.
The GCM is also suitable for situations where the focus and/or zoom can vary, as will be described in Section 3.1.

3. Calibration

To use a camera model its intrinsic parameters have to be known, and these are determined in a calibration. That can be done by taking several images of reference points from different angles and distances, and perform a calibration calculation based on the images. In these calculations the positions of the references will be calculated, as well as the camera poses for the calibration images. The reference positions are also useful if there is a need to calculate camera poses based on images, described in Section 4.1. The calibration problem can be transformed into a least squares optimization problem. If the optimization is made in image space the function to minimize is obtained from the norm of a residual vector of distances between the measured image points x_c^i and the estimated points, $\hat{x}_c^i$ calculated based on (24) of the camera model and its partly unknown parameters:

$$\min \sum_j \sum_k \left| x_{cjk}^i - \hat{x}_{cjk}^i \right|^2 \tag{29}$$

The sum is taken over all calibration images with indices j and all reference points with indices k. It is an advantage if approximate initial values of reference positions, camera poses and intrinsic camera parameters are known. Otherwise the optimization procedure may not converge, see Section 3.3.
One optimization issue is that it has to be known which reference point in the object world corresponds to which point in the images, the correspondence problem. One way of solving that is to place unique groups of references in the environment, like star constellations. These can be matched with a matching algorithm. Actually the references together can be viewed as one large constellation, but then it takes a longer time to match. Another way of doing this is to manually instruct the system the identity of each point in the images. The same optimization criterion, (29), is used weather the reference positions are known or not, but if they are not known they are calculated by the optimization program, otherwise they are considered given constants.
In the calibration calculations there are a large number of unknown parameters. If the references are unknown there are six pose parameters for each calibration image, three for each reference position in addition to the intrinsic camera parameters. A residual vector containing all the optimization image errors can be sent to the optimizer. The optimizer can then square and sum if needed. Also a Jacobian matrix can be sent to the optimizer. This Jacobian contains all the partial derivatives of all the elements in the residual vector with respect to all the unknown parameters. Calculation time can be saved by using the fact that

most of the elements in this matrix are always zero. For example the derivative of a residual element corresponding to one image is zero with respect to a pose parameter of another image camera pose. The same is valid for the derivative of an image reference position with respect to the position of another 3D reference point. The derivative with respect to an unknown intrinsic camera parameter will in general not be zero though. These derivatives can be computed numerically, or analytically. The residual vector can have the double length if each image point difference is divided into individual difference in x and y direction.

The optimization can also be made in object space. By using a measured image reference coordinate, the camera model can be used to calculate the corresponding object line $x_{fo}^w + a(x_{fi}^w - x_r^w)$. The shortest distance from this line to its corresponding 3D reference point position x_o^w can then be used as residual. An optimization criterion for this case is

$$\min \sum_j \sum_k \left(\frac{\left| (x_{fijk}^w - x_{rjk}^w) \times (x_{fojk}^w - x_{ok}^w) \right|}{\left| x_{fijk}^w - x_{rjk}^w \right|} \right)^2 \tag{30}$$

In the parenthesis is the formula for the shortest distance between a point and a line. Minimization can also be performed at any stage in between, for example the points can be converted to non-decentring distorted image coordinates and the residual vector is formulated accordingly. Yet another possibility is to use (28), and minimization can be done in a non distorted image plane, if f_{outer} is constant. There are calibration methods specialized in minimizing errors in a non distorted image plane, that uses the fact that there straight lines are mapped to straight lines in the images (Devernay & Faugeras 2001), and these can be applied also for the GCM if (28) is used. Reference points can be placed in the environment for that purpose, or natural points and corners can be used.

Another way of performing the calibration when camera poses can be measured independently, e.g. by a laser tracker, is to also include differences between the calculated calibration poses from vision and measured poses in the optimization criterion. If the calibration camera poses are measured, the calibration is called active. One thing to consider then is that if the systems measure in different coordinate systems, and it is not exactly known which point on the camera is measured transformation parameters will be added as variables in the optimization. Weights should be added to the least squares error components, to get the right importance of image pixel errors compared to position errors and orientation errors.

3.1 Variable Focus and Zoom

The geometry of the camera has been considered constant so far. It is possible to allow for variable focus and zoom in the system using the GCM if there is some kind of measure of how the camera is focused and zoomed. There are two reasons that the GCM is suitable for variable focus and zoom applications. One is that the position of the camera can be a point on the detector, and not in the lens system that can move when zooming and focusing. Another is that the entrance pupil can move considerably when changing the zoom.

The calibration procedure would be similar as for fixed geometry, but the calibration images must of course be taken with different focus and zoom. Some kind of interpolation or function dependency will be needed between different values of the focus and zoom parameters and the other intrinsic camera parameters. If f_{oc} is the focus and z_o is the zoom, one way is to let the intrinsic parameters depend on them in the following way

$$c(f_{oc}, z_o) = q_0 + q_1 f_{oc} + q_2 z_0 + q_3 f_{oc}^2 + q_4 z_o^2 + q_5 f_{oc} z_o \quad (31)$$

That is each camera parameter can be replaced by this kind of expression. This implies there will be six times as many camera parameters. These dependencies are in general different for different camera parameters though, e.g. m_1, m_2 and s for the image plane conversions do not depend on the focus and zoom and therefore do not imply more parameters. Another way of calculating with variable focus and zoom is to do a regular calibration for some fixed values of these parameters, and then use interpolation between them, like in Fig 5. Here a triangular net is constructed in the 2D space of f_{oc} and z_o values.

If the system is calibrated in the points of Fig 5, then if for some value of these parameters it can be determined which triangle we are in, e.g a linear interpolation between the corner points of the triangle can be done. If a projection to the image shall be done, first calculate the corresponding image coordinates for the triangle corners, and do the interpolation. A similar thing can be done in the other direction by calculating the vectors of the object line and do a linear interpolation. If it is important to have exactly the same transformation both from image to object space and vice versa, as in the method to be shown in the last part of Section 3.3, the following can be done. Make an interpolation of the different camera parameter values in the triangle corners, and use them to do the camera model projections. A rectangular net can also be used, then a bilinear interpolation should be used.

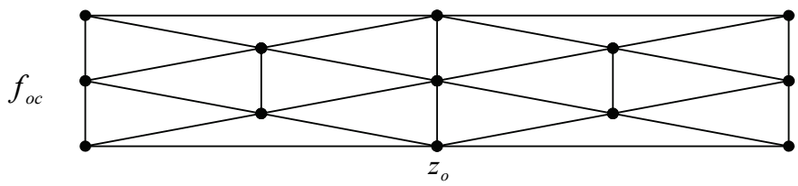

Fig. 5. Triangular net used for interpolation between points calibrated for different values of focus and zoom.

3.2 Nontrivial null spaces

Before the actual calibration calculations, we consider so called non trivial null spaces. These are ambiguities occurring when the same image can be formed with different parameter setups, such as camera parameters and camera poses. That can cause problems when the calibration is based on images. One obvious null space occurs when changing the world coordinate system. By redefining the world coordinate system or moving the complete system with camera and references the same images can be obtained. Therefore the coordinate system should be locked somehow before the calculations.

Seven parameters need to be locked for defining the world coordinate system, three for position, three for orientation and one for scaling.

First force the system to have the coordinate system origin, $(0,0,0)^T$, in one reference point. Let the x-axis be in the direction to one other reference point by measuring the distance, x, between these points and set the coordinates of this second reference to $(x,0,0)^T$.

Then the z- coordinate of a third point can be set to zero, defining the final rotation degree of freedom. These numbers should be constants, and hence not be changed during the calibration calculation. The length scales of the system will be defined by x. The more exact x is measured, the more accurate the length scales will be. This will define a 3D world coordinate system that the vision system relates to.

Another nontrivial null space occurs when determining the constant e_0 of f_{outer} in (8). If it is increased for a camera pose, the same image can be formed by in the same time moving the

camera backwards the distance of change of e_0. To solve that e_0 should be set to a constant and not be changed during calibration, if focus and zoom are constant. As a matter of fact it is usually best to set that coefficient to zero. That will give a camera model that differs a bit from Fig 2, but mostly for pedagogical reasons it was first described it in that way. Instead in Fig 2 the point x_{f0} together with the object line will be moved down close to the image plane (or more correctly the image plane and x_{fi} moved up). This makes the position of the camera defined as a point in the centre of the optics, just as with the CCM, which is good. One exception of e_0=0 is if the camera is calibrated also with respect to varying focus and zoom. Then e_0 can be a function of these parameters, but to find that function dependency the calibration camera positions have to be measured with some other system, as in the last part of the calibration section, because of this null space. One way of determining the dependence of e_0 with respect to zoom is to look at the camera and see how much the front lens moves when zooming. That dependency can be used for that parameter and then optimize the rest with respect to images.

Another null space is in the values of the constants m_1 and m_2. Their ratio, the aspect ratio, is the ratio between the pixel distances in x- and y- on the detector. Though one of them can be set to any positive value. Therefore it is convenient to set one of them to one. Then the unit of f_{inner} will be a pixel distance unit. The unit of f_{outer} is the same as for an outer world coordinate system. The other of the two constants m_i can be seen as a camera parameter, to be determined in the calibration procedure. The calibration will then find it to be the mentioned ratio between the pixel distance in the image directions. With this method we actually don't have to know the actual pixel distances in the detector, in fact it can not be determined in this kind of calibration, and it is not needed, at least for the applications here.

Other nontrivial null spaces can occur when the detector surface is parallel to a planar reference plate. Consider, for simplicity, a PCM. Then if the camera is moved away from the reference plane and in the same time the focal distance is increased, with the same proportions as the movement distance from the plate. Then the same image would be obtained, see the left part of Fig 6. In the figure two "light beams" are shown but actually the whole image will be the same. This can cause problems if the calibration images are non-leaning enough compared to the reference plate, even in calibration with known references. Therefore images with leaning camera have to be included in the calibration, at least in passive calibration and the references are in a plane. This can occur also when the camera has distortion.

If the detector is not perpendicular to the optical axis, and again the references are in a plane another null space can occur. Think again of a PCM, but with leaning detector compensation. Then the detector can be kept parallel to the reference plane even though the optical axis is leaned, Fig 6. If this camera is moved to the side, and the lens system is held directed towards the same point on the reference plate (by leaning the detector surface), then the image will not change. Hence this is another nullspace. This is another reason to include calibration images with leaning camera poses.

In order to get a good calibration the calibration images have to be such that the references seen in the images cover the range of inclination angles, α, the system shall be used for. If the entrance pupil is not constant compared to the camera the calibration images have to vary both in distance and angle to the references. Otherwise the calibration calculation can make the system work accurately only for the particular distance of the calibration images, see Fig 7. The problem occurs if there is a large enough interval of α where the distance to

the references does not vary. That is because of another null space. If somewhere in the α interval f_{outer} is e.g. too large that can be compensated by having f_{inner} a little too small, see Fig 7. Then the system will be accurate only for one distance in that direction, α .

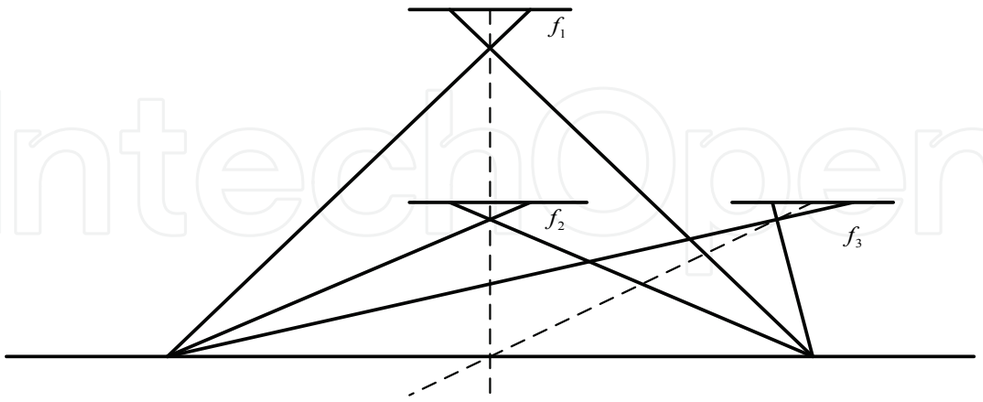

Fig. 6. If the detector surface is parallel to a reference plane nontrivial null spaces occur.

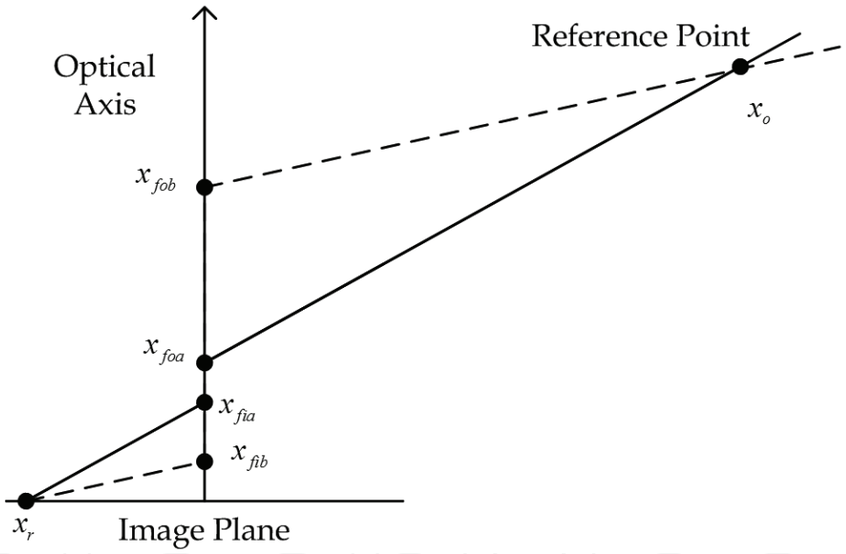

Fig. 7. If the distance to the reference does not vary two different sets of camera parameters *a* and *b* can both give a small error norm in both image and object space. The calibration can not determine if situation *a* or the dotted *b* in the figure is the right one.

3.3 Pre-processing algorithms

This section suggests means for improved calculation results, especially initial values for extrinsic and intrinsic camera parameters as well as reference positions to assure convergence, and estimation of the centre of a reference in the image. The centres of the references are important since the centre of gravity of a projected shape in the image does in general not reflect the centre of the object in object space. The calibration is a large calculation, and without estimations of the parameters it is hard for them to converge at all. For the methods in this section to be valid the references needs to be flat and with a known geometry.

Estimating Intrinsic Camera Parameters

Initial values of intrinsic camera parameters are estimated based on the shapes of images of reference points. For each reference in the image, extract image coordinates around its border on a sub pixel level by using greyscale intensities of the pixels. Use the camera model to convert these points to object space lines, see Fig 8, project them to a plane in object space, and then match the known real shape and size of the reference to these object lines using optimisation routines. The set of intrinsic parameters that give best match are used as initial estimates of the camera parameters.

This can be done to several or all of the references in the images. When a good match is achieved we have probably found good estimates of the intrinsic camera parameters.

Details in the problem formulation are now given, and to ease the work start the calculations in the camera coordinate system. The points used in the camera model to project to the object space is simplified to

$$x_{fi}^c = \begin{bmatrix} 0 \\ 0 \\ f_{inner}(r) \end{bmatrix}; \quad x_{fo}^c = \begin{bmatrix} 0 \\ 0 \\ f_{outer}(r) \end{bmatrix}; \quad x_r^c = \begin{bmatrix} x_{r1}^i \\ x_{r2}^i \\ 0 \end{bmatrix} \qquad \text{(32 a,b,c)}$$

The corresponding object line is then expressed by inserting (32) into (13). The equation of a plane through an estimated flat reference can be written

$$\text{Reference Plane: } \hat{x}_o + a_1 v_x + a_2 v_y \qquad (33)$$

The vectors $\hat{x}_o$, v_x and v_y define the location and angle of the estimated reference plane and a_1 and a_2 are parameters.

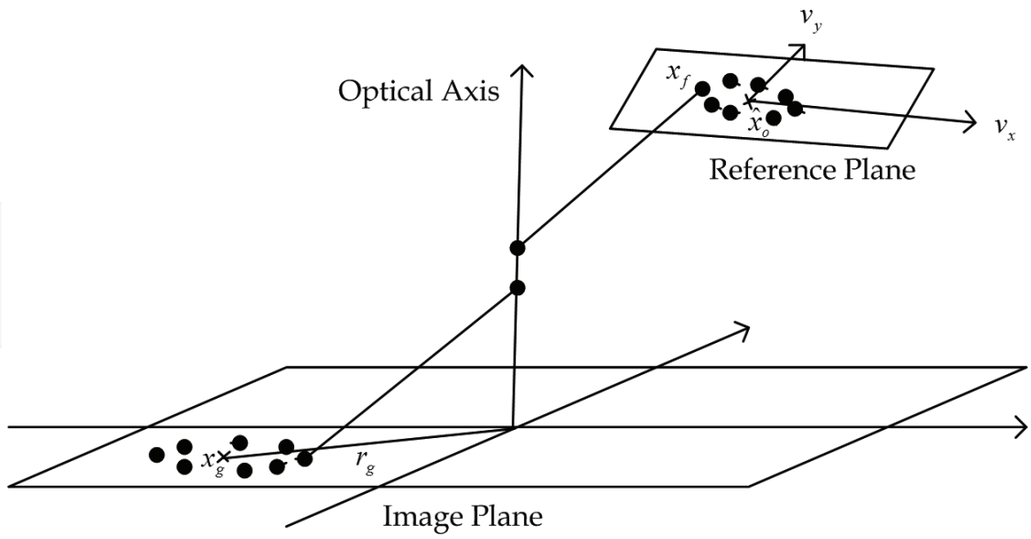

Fig. 8. For estimating calibration parameters, and also calculating reference image centre points, the boundary of the image of the reference is projected to a reference plane in object space, the results are points x_f. $\hat{x}_o$ is the centre of the points x_f. After optimization of (34), $\hat{x}_o^c$ will be estimations of reference positions x_o^c.

The vectors $\hat{x}_o$, v_x and v_y are first estimated roughly, see below, and then an optimization procedure can be used to adjust them more exactly. By setting the formula for the plane (33) equal to the object line expression, a system of equations is obtained where the solution, x_f, is their intersection. Without loss of generality $\hat{x}_o$ can be set as the centre, or some other well defined point on the reference, in the object world. v_x and v_y span the plane, and must not be parallel.

Now the least squares sum can be expressed. Index j represents the camera poses, index k the reference number and l is a numbering of the points around the image of a reference circle. The sum should be minimized with respect to the reference centre points and the plane angles ($\hat{x}_o$, v_x and v_y) and the intrinsic camera parameters. For circles this can be expressed as

$$\min \sum_j \sum_k \sum_l (\left| x^c_{fjkl} - \hat{x}^c_{ojk} \right| - r_k)^2 \tag{34}$$

This is a sum over squared differences between the distances from the out projected border points x^c_{fjkl} to the estimated reference circle centre points, $\hat{x}^c_{ojk}$, and the circle radius, r_k. The points x_{fjkl} are assumed to be in the projection planes, which they always are if they are calculated as the intersection between the plane and the lines described above. The radius r_k of the reference points should be measured before the calculations, so that they are known and inserted in the expression. If the radius is not correctly measured it will change the distance between the camera and the reference points in the calculations, but the values of the intrinsic camera parameters will be almost exactly the same. The calculations can also be performed in two steps so that an outer optimization of the camera parameters calls a plane matching optimization routine. This should give more stable calculations.

Since this is only an estimation calculation decentring distortion should probably be ignored, as well as the entrance pupil variations. As starting approximation of the principal point, the centre of the detector can be used. This coordinate can be further optimized if needed.

Computationally it is efficient to just start with a few references, and add more while the estimation is improved. When adding new references starting values of $\hat{x}_o$, v_x and v_y should be estimated for the reference. From the centre of gravity x^c_g of the image coordinates around the reference an object line can be computed that approximately should go through the reference x^c_o. But at first we don't know the distance from the camera. To determine the distance a test plane can be constructed somewhere on the object line that is orthogonal to it. Calculate object lines based on each image point around the reference, and calculate their intersections with the test plane. The test plane is obtained by setting the vectors (35) into (33).

$$\hat{x}^c_o{}' = x^c_{fi}(r_g) - x^i_{g1} e^c_x - x^i_{g2} e^c_y + x^c_{fo}(r_g) \tag{35a}$$

$$v^c_x{}' = e^c_z \times (x^i_{g1} e^c_x + x^i_{g2} e^c_y) \tag{35b}$$

$$v_y^c{}' = (\hat{x}_o^c{}' - x_{fo}^c(r_g)) \times v_x^c{}' \tag{35c}$$

The intersection can be calculated with linear equation systems, setting (33) equal to (13). The unit vectors in (35) in the camera coordinate system are simply $e_x^c = (1,0,0)^T$, $e_y^c = (0,1,0)^T$ and $e_{cz}^c = (0,0,1)^T$. From the out projected points not only the direction but also the distance and the angles can be compued. Calculate the out projected points centre of gravity, x_{gf}^c, in the plane defined by (35). Find the out projected point with the maximum distance to this centre of gravity. Call a vector from the centre of gravity x_{gf}^c to the point furthest away $R_{\max}$ and the corresponding distance $r_{\max}$. Find out the shortest distance to the centre of gravity, call the distance $r_{\min}$. Observe that $r_{\max}$ is the radius corresponding to a non-leaning direction of the circular reference. Therefore the ratio between r_k and $r_{\max}$ determines how far from $x_{fo}(r_g)$ the reference is, measured in units of $\left|x_{gf}^c - x_{fo}^c(r_g)\right|$. In the direction corresponding to $r_{\min}$ the reference plane leans the most corresponding to the test plane. The leaning angle in that direction is $\arccos(r_{\min}/r_{\max})$. Good starting values for $\hat{x}_o$, v_x and v_y in the optimization (34) can now be expressed as:

$$\hat{x}_o = x_{fo}^c(r_g) + \frac{r_k}{r_{\max}}(x_{gf}^c - x_{fo}^c(r_g)) \tag{36a}$$

$$v_x = v_{hc}\cos\chi + v_{ha}\sin\chi \tag{36b}$$

$$v_y = v_{ha}\cos\eta \pm v_{hb}\sin\eta \tag{36c}$$

where the vectors v_h are

$$v_{hb} = \frac{x_{gf}^c - x_{fo}^c(r_{cg})}{\left|x_{gf}^c - x_{fo}^c(r_g)\right|} \tag{37a}$$

$$v_{hc} = \frac{R_{\max} - x_{gf}^c}{\left|R_{\max} - x_{gf}^c\right|} \tag{37b}$$

$$v_{ha} = v_{hb} \times v_{hc} \tag{37c}$$

v_x and v_y contains angles. It is usually better to adjust these angles in the optimizations (34) than the vectors v_x and v_y directly. Starting values of the angles are then 0 for χ and $\arccos(r_{\min}/r_{\max})$ for η. There are two values of v_y because of the plus and minus signs. Both should be tried, and the one leading to the best fit is used. The calculation of starting values for intrinsic camera parameters here are not only used for that, but also as a starting values in the next two sections.

Approximate Reference Positions and Pose Calculations

In the above calculations not only the intrinsic camera parameters were estimated, but also the relative positions of the references and the camera poses, through the $\hat{x}_o^c$ vectors.

Therefore from the output of the optimization in (34) the relative positions of the reference positions, and also the different poses for the calibration images, can be approximately calculated also in the world coordinate system. These positions are valuable starting points for the calibration procedure.
The extrinsic camera parameters and reference positions are calculated by matching the reference positions that are now given in the camera coordinate systems for the different calibration poses. The idea is to transform the systems by rotation and translation so that the reference positions match each other as well as possible.
One way of making the coordinate transformations to calculate camera poses and reference positions, is again to optimize using a least squares expression.
The relative positions of the reference point to the camera positions are $\hat{x}^c_{ojk}$, calculated in the initial camera parameter estimation section above. These are transformed by a rotation and translation for each image, similar to (1) but this time from camera to world coordinates, to get the points

$$\hat{x}^w_{ojk} = R_j \hat{x}^c_{ojk} + T_j \tag{38}$$

Here $\hat{x}^w_{ojk}$ are approximations of the reference point positions x^w_{ok}. These are transformed for each image *j* so that the different approximations of the same point match each others as well as possible. R_j is a rotation matrix defining rotations by three angles around the three axes, T_j is a translation vector with three components. These are transformations from the different camera coordinate systems to the world coordinate system. The least squares expression should be optimized with respect to T_j and the three angles in R_j and can be expressed as:

$$\min \sum_k \sum_j \left| (\hat{x}^w_{ojk} - \frac{1}{n_k} (\sum_i \hat{x}^w_{oik})) \right|^2 \tag{39}$$

This is the sum of squares of the reference points distances to their average position based on all images. The sum over *i* is actually another sum over images where n_k is the number of images for each reference point.
Here some degrees of freedom need to be locked, to prevent the optimization from "drifting away". One way of doing that is to formulate a variation of (39). For the three references with fixed coordinates for the calibration (see Section 3.2), replace the aveage sum for these reference parameters with the known fixed values. The optimization in (39) now gives all the elements in every R_j and T_j which hold the information of the poses of the camera for the images j.
As approximation of the reference positions the last sum in the expression (39) can be used. If references are already known before the calibration calculation a similar method can be used to calculate the calibration camera poses R_j and T_j, but then the last averaging sum should be replaced by the known reference positions.

Image Reference Coordinates

Image points that correspond to certain points in the 3D object world are needed in the calculations. If we for simplicity assume circular references, it is natural to use the centre of

it in the 3D object world as the reference 3D position. To get a good accuracy we need to know a 2D point in the image that corresponds to that 3D point. It is not optimal to use the centre of gravity as the reference point in the image, not even if a simple PCM camera is used. What we want is an image point that as exactly as possible is "looking at" the reference centre point. The result of (34) is a good starting point also for calculating these image centre points. The image references are projected to planes in the object space in (34), and their 3D positions of their centers are calculated. Now we can just project these points back to the image using the camera model. We even know the reference points in the camera coordinate systems, giving even simpler projections, since equations (15,16) can be used right away without doing coordinate transformations first. The reference positions and camera parameters are only approximately known, but when projecting the centre point back the errors will cancel each other. That is since the error in projecting to the object space will be done in reverse when projecting the centre point back, giving good image points. Another method for centre points of circular references can be found in (Heikkila 2000).
These centre point calculations can be made both before the calibration and in pose calculations that will be explained in Section 4.1. Good values of the image reference positions are important for accuracy of calibration and pose calculations, and also give better defined optimums in the algorithms making them converge easier and faster. There is a trade off between calculation time and accuracy. If accuracy is important, new image reference coordinates can be calculated again using preliminary camera parameters, and then calibrate again. In that case, when the new centre points are calculated the camera parameters should be kept constant to the values from the previous calibration when calculating new centre points, but be optimized again in the new calibration.

3.4 Further improvements of calibration

The accuracy of this kind of vision system has been measured using a Coordinate Measuring Machine (CMM) as a reference. The system was calibrated, and then pose calculations were made based on the calibration, and finally the calculated poses were compared to CMM position data. The pose calculations will be described in Section 4.1.
After the camera was calibrated it was placed on the hand of the CMM. The CMM was programmed to move to different locations above a reference plate with reference IR-LEDs. Measurements with both the vision system and the CMM were done in the locations where the CMM stopped. The two systems measured in different coordinate systems. Therefore, to do the comparison the coordinate systems had to be matched, so the vision coordinates were transformed into the CMM coordinate system. The transformation was done with a matching procedure, where seven transformation parameters were determined. The parameters were three for rotation, three for translation and one for scaling. They were determined with an optimization program. One part of the CMM positions was scanning along a line vertically, away from the centre of the reference plate, stopping every centimetre along that line. A plot of the error in mm in the z- direction, the direction away from the reference plate, along that line can be seen in Fig 9. The left diagram is based on an ordinary calibration for the GCM as described above.
The curve is almost straight but leaning. A good result would be a curve close to the $y = 0$ line. It seems like some systematic error is involved. This behaviour occurs when the distance error in the direction to the reference plate is proportional to the distance to the reference plate. That is what happens when the focal distance is wrong, or the f_{inner}

polynomial is wrong in a way similar to multiplying it with a constant. That can happen because of the nulls pace shown in Fig 6. It shows that there were not enough leaning calibration poses. To correct that error, first the first constant in the f_{inner} polynomial was locked to a slightly lower value, but the rest was calibrated as before. The corresponding curve after that is the centre curve of Fig 9. The curve is better since the errors are smaller. It is not leaning much, but has been a little more bended. That the curve does not just lean down but gets bended indicates there are some systematic bias that wants to change f_{inner}.

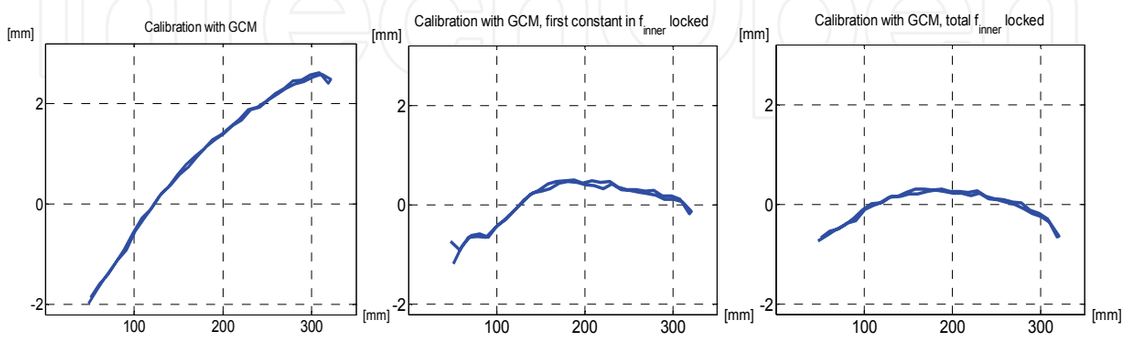

Fig 9. Error curves in mm from scanning a line in direction away from the reference plate. Only the errors in the direction away from the reference plate are shown. Four radial distortion parameters are used. Left, ordinary calibration from images using GCM. Centre, curve leaned down by adjusting the first constant in $f_{inner}(r)$. Right, the whole polynomial $f_{inner}(r)$ multiplied with a constant.

The residual is not only the same (as in the null-space) when changing f but it strives to be a slightly wrong value. The reason can be that because of how the diodes are mounted on the reference plate, leaning camera angles sometimes can not see the whole reference diode, a part of it is shadowed. Therefore the leaning images that should correct this error do not have perfect image coordinates. Also the light intensities from the diodes are lower for larger angles, making them harder to see from the side. Another way of trying to correct this is to not only change the first parameter in f_{inner}, but multiply the whole polynomial after calibration with a constant. That can be achieved by multiplying all the parameters in f_{inner} with a constant. So the system was calibrated as usual and then the f_{inner} parameters were multiplied by another constant, and then the system was calibrated again but with f_{inner} parameters fixed. Then the resulting diagram was the right one in Fig 9. This is a little better than the centre curve, and the procedure give a slightly smaller errors (smaller error norm in mm, see Section 5). The same calculations were done with the CCM. The result of ordinary calibration was almost the same as the left diagram in Fig 9 for the GCM. This can be corrected for the CCM in a similar way as the GCM was corrected.

Note that this behaviour of leaning z- curve would not be possible to see if only a line is scanned in the z-direction. That is since then the coordinate matching optimization would take that error away. The poses have to be spread parallel to the reference plate also, when the coordinate transformation between the vision coordinate system and the CMM coordinate system is done.

The procedure of including measured poses in the calibration could also correct this error. That procedure could not be done in a good way here because once the camera was placed on the CMM it could not rotate. The angle should vary for the different calibration images.
If enough non biased leaning calibration images are used this error should disappear. So this correction is not always necessary. It could sometimes be a good idea to do this control of comparing with another position system, like a CMM, to be sure that the calibration is correct.

4. Applications

Methods for using the calibrated camera model for calculating a camera pose based on an image is first shown here. Then a method of using stereo vision from calibrated cameras for calculating positions is described.

4.1 Pose calculations

The 6D pose of a calibrated camera taking an image can be calculated using a single image. That is done in a similar way as the calibration, but both the camera parameters as well as reference positions have to be known, as they are after the calibration. Project the reference points to the image using the camera model, and calculate the difference to the detected image points. In that way an optimization criterion can be formulated to be solved by an optimization program. Also, the references need to be recognized somehow, to know which reference in the object world is which in the image. The same error criterion can be used as in the calibration, but the camera parameters and the reference positions are known constants. The only things to adjust by the optimization program are the six pose parameters. Since only one image is used there is no summing over images here. To calculate the pose from an image it needs to see at least four references. There will be the same number of unknowns as equations for three references, but there will still be many solution points, so at least four should be used. A better accuracy can be achieved with more references. Just like in the calibration an object space minimization criterion can be used instead.
These methods can also be used if there is a need to know the relative pose between a camera and an object with known geometry, known from e.g. a CAD model. First extract corners and edges of the object in the image. Then try to match it with the CAD model by projecting it to the image with the camera model. An optimization program can find the relative pose that best matches the image.
A similar procedure can be used in object recognition. If a set of known 3D geometries can appear in the image, then try to match the image with a mathematically projected image calculated with the camera model and the known geometry. If an optimization can find a good match the object is found.

4.2 Stereo vision and other 3D vision methods

To find the position of a point using vision normally at least two viewing angles are needed in order to find depth, if there is no other known geometrical information about the viewed point. To use stereovision, take images of the point or points of interest from (at least) two directions. Then find the same point in both of the 2D images. The camera poses need to be known e.g. with the method in Section 4.1, and the camera needs to be calibrated. Then

calculate the two corresponding object lines using the camera model. The crossing of these lines is then the position of the observed point. Since the system is not exact the two lines will probably not cross each other exactly. But the closest points of the two lines can be calculated. Assume we have two object lines, one for each camera view, calculated from (13) or (14)

$$\text{Line 1: } x_{0a} + av_a \text{; Line 2: } x_{0b} + bv_b \tag{40a,b}$$

An equation for the points closest to each other on the two lines are

$$\begin{bmatrix} v_a^T v_a & -v_a^T v_b \\ v_a^T v_b & -v_b^T v_b \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -v_a^T x_{0b} + v_a^T x_{0a} \\ -v_b^T x_{0b} + v_b^T x_{0a} \end{bmatrix} \tag{41}$$

This equation can be derived from the fact that a straight line between the points on the lines closest to each other is perpendicular to both of the lines. The equation can be solved for the parameters a and b. When these are inserted into (40) the points on the lines that are closest to each other are obtained. The centre point between these points can be used as the 3D position searched for. The distance between these points on the lines can give a hint of the accuracy, especially if several points are calculated or if a similar procedure is done from more than two images of the same point. If a point is found in one of the images of which the 3D coordinates are wanted, then its object line from that point can be computed and projected to the other image(s), for different values of the parameter. Then a line in the other image is obtained where the point should lie. This makes it easier to find the point in the other image(s).

If there is a known constraint on the point, e.g. the point is on a known surface, then the coordinates of the point can be calculated from only one image. The coordinates of the point will be the intersection between the object line and the known plane or line.

Structured light is another method to determine 3D coordinates (Zhou & Zhang 2005). If a point is illuminated with structured light and the position of a light line is known the position of the point is the intersection of the object line and the light line. Another way of using structured light is to have a light source that projects a line, that is the structured light is in a plane. If the geometry of the plane is known the 3D geometry of a line can be calculated. First find the 2D line in the image corresponding to where the light plane hits an object. Then several object lines can be calculated, one for each image pixel of the line in the image. Points following the 3D curve can be calculated as the intersections between the object lines and the structured light plane. This method to find the curve will not work if the camera is in the same plane as or too close to the structured light plane.

Stereovision can also be used for determining the 3D geometry of a curve. Take images of the curve from two different directions. Find the curve in both of the 2D images. Sort the points according to position along the curve. Then go through the points of one of the images and calculate the "object line" for the point. By going through the points of the other image find which of their object lines are closest "above" and "below" the first object line, using (40-41). In that procedure the following expression is useful

$$(x_a - x_b)^T (v_a \times v_b) \tag{42}$$

where x_a and x_b are the points on the lines (40) obtained by using (41). (42) will have a different sign for points "above" and "below" the first object line (40a), so it can be used for determining the closest "object lines" (if v_a and v_b always points in the directions from the

camera to what is observed or always in the opposite direction, which is the case in practice). Once the closest object lines from the other image are found use their closest points on the first object line and determine their average position. The average should be weighted inversely proportional to the distance to the lines from the other image. The result can be used as a 3D point that is on the 3D curve searched for. By repeating this with the image points along the curve in the image a sequence of 3D points will be obtained along the 3D curve. Some method of smoothing the curve can be used if needed. An alternative here is to make a plane of the object line closest "above" and "below", and determine the planes intersection with the first object line (calculated by setting the object line equal to the parameterised plane). The plane is well defined if the object lines are determined from a camera with constant f_{outer}. The fact that the distance between object lines is given in a straightforward way with this method (40-41), and also on which sides the lines are compared to each other (42), is useful here. Therefore the method is suitable for determining curves as well as points.

In doing these stereo calculations, as mentioned, the two different camera positions needs to differ to get a good accuracy, and to be able to do this at all. When the geometry of a curve is determined in this way the camera poses have to differ in a way that the cross product of unit vectors along the object lines has as big component along the observed 3D curve as possible. If the component of this cross product along the curve is zero it is not possible to use this procedure, without being able to distinguish each individual point along the curve. That is, then it is not enough to know that the two curves correspond to each other but also which point corresponds to each other.

Sometimes there is a need to know relative positions between points or curves both seen in the stereovision images. Then if one of them is already known there could still be a good idea to determine the positions of both of them with the vision system. Then the difference between the points calculated with vision, from the same images, can be used as their relative positions. This can increase accuracy since approximately the same errors can occur for both points, so when the difference is calculated, this error will almost cancel each other out. This is for example the case if stereovision errors originate from not exactly known camera poses.

5. Accuracy

To compare accuracy the calibration and pose calculations were implemented in MatLab. The calculated poses were compared with data from a Coordinate Measurement Machine (CMM). Both the GCM and a CCM were used, with the ability to choose the number of and combination of camera parameters, and hence different kinds of distortion accounted for in the model. Both the residual vector and Jacobian were calculated and sent to the optimization program. The Jacobian was calculated numerically, but the components that are always zero were not updated in the program, to increase the calculation speed. Calibration images and images taken with the camera placed on the CMM together with corresponding position data from the CMM were input data to the calculations. The calibration images were photos of a reference plate from different positions and angles. As mentioned in section 3.4, the vision coordinate system was transformed into the CMM coordinate system before the poses were compared. The seven parameters of this transformation were calculated with an optimization program. Only position errors were compared and not orientation errors.

As references, IR-LEDs were used, mounted on the reference plate. These references were grouped into twenty unique patterns of six diodes, to solve the correspondence problem. To make the calculations faster only one reference per pattern were used in the calibrations and pose calculations, except for two of the patterns where two diodes were used. They helped defining the coordinate system. The calibrations, with the different variations of models and distortion parameters, calculated intrinsic and extrinsic camera parameters as well as the reference positions. An image space minimization criterion was used in the calibration, as well as in the pose calculations. The CMM scanned two lines parallel to the reference plate, at height five and nine centimetres, and one line perpendicular to the reference plate, above the centre of it. Each scanning stopped at every centimetre along the line, and each line were scanned twice. To analyse the accuracy, average differences between the vision system positions and the CMM positions were used, together with a residual vector error norm after the calibration optimization. The residual vector contains image errors measured in pixel units. The camera had 640*480 pixels. The angle of view was 60 degrees in the x-direction and a bit less in y. 526 calibration images were used, the same images for every calibration. In comparison with the CMM, poses calculated from seven or more references were used. The focus and zoom are the same for all the images.

Errors results can be shown with error diagrams, similar to Fig 9. To compare different combinations of camera parameters and the different models, bar charts of error norms are made in Fig. 10 and 11. First a bar chart for the GCM with only radial distortion and constant entrance pupil is shown, with different degrees of the f_{inner} polynomial (7). The principal point and the aspect ratio are held constant. The parameter s is zero. The average error in mm in 3D is shown for the three lines scanned above the reference plate. The improvement in Section 3.4 is used for the errors in mm. Also the error residual norm from the calibration is shown, this is an image pixel error norm, and this is based on an optimization not using the procedure in 3.4. In the table it can be seen how much is gained from adding more parameters for radial distortion. The error is around 0,35mm in 3D position or 3 pixels in the 2D image when two or more distortion parameters are used. Possible error sources when many radial distortion parameters are used could be bad quality of calibration images, as discussed above, not regular detector pixel coordinate system, bad focus, not exact centre points of the image of the references. Another source of error could be that the object line is not exactly straight especially for close ranges.

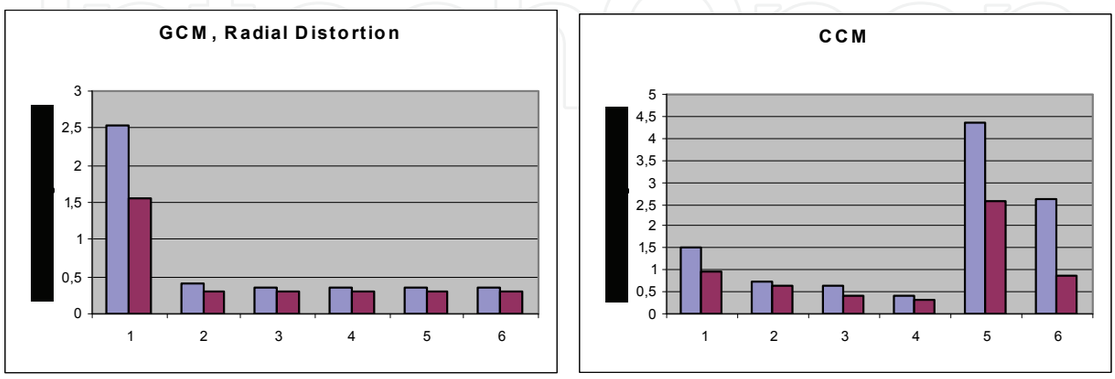

Fig 10. Bar charts over error norms. Left bar chart shows GCM with only radial distortion. The x- axis shows the number of distortion parameters. The first of the double charts shows

average position error in mm in 3D for the calculated poses, the second is an image pixel error norm after calibration. The pixel errors are divided by ten to be able to use one scale in the *y*- axis. The right bar chart shows errors with the CCM. In the first four bar pairs the x-axis shows number of radial distortion parameters. The second last pair has one quadratic distortion term, k_{p2} in (3c), and the last has one quadratic, k_{p2} , and one with power four, k_{p4}.

For a PCM, i.e. the GCM model with only one parameter in f_{inner} and no distortion parameter, the errors are about 13 mm and 6,6*10 pixels. It is not shown since it would give a too small scale in the diagram. If the procedure in Section 3.4 was not used the error norm in mm would be 1.14 instead of 0.35 for four radial distortion parameters with the GCM. A similar gain was achieved for the CCM.

The corresponding bar chart for the CCM is shown in the right bar chart of Fig 10, for one to four radial distortion parameters. First both odd and even powers in the radial distortion of CCM were used. Then it was investigated what happens when only even powers are used. The procedure of Section 3.4 was used to improve the accuracy in mm, even though it can be viewed as a method for the GCM. If that was not used the errors in mm would be considerably larger, see above. Since image minimization is done it is easier to use the type of CCM conversion in (5) instead of (6), so (5) is chosen.

By comparing error norms between the GCM and the CCM with the same number of distortion parameters, it can be seen that with only one distortion parameter the accuracy is better with the CCM, but when more parameters are included the GCM has a better performance. It can also be seen that a better result is actually achieved when including also odd powers in the polynomial (5), even though most of the CCM papers suggest only even powers. This is perhaps not true if the polynomial is used in the other direction as in (6), in that case it can be an advantage to use only even powers.

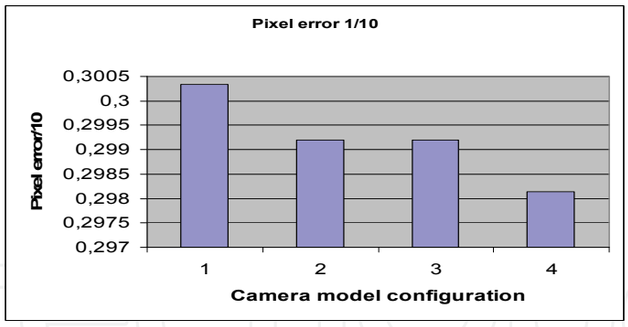

Fig 11. Calibration residual norm in pixels, again divided by ten. Results for the GCM, all four have four radial distortion parameters. The second has also leaning detector compensation with constant $f_l(r)$. The third has varying entrance pupil with a second degree $f_{outer}(r)$. The fourth has both leaning detector compensation and varying entrance pupil.

When adding the additional distortion types the results can be seen in Fig 11. Here the GCM is used for the ordinary radial distortion and then other distortions are added. For ordinary radial distortion four parameters were used. Here only the calibration error residual norm was used for comparison. This is because for the camera used, the improvements of more parameters were small. Therefore since a "manual part" described in Section 3.4 is included

this does not give an exact enough comparison. So the residual vector norm is probably a more exact measure for how much can be gained with the additional types of distortion in case better leaning calibration images were included in the calibration. Here it can be seen that with this camera very little was gained when using decentring distortion and varying entrance pupil. Decentring distortion was not important because the particular camera used did not have much decentring distortion, probably it can be important for other cameras. Also entrance pupil variations did not give a much higher accuracy with the camera used. If it had a larger viewing angle and also a larger lens, that could be more important.
For decentring distortion only the leaning detector formula (18) was used and not (20-23). Since the leaning detector angle was very small (as well as the gain in residual norm), probably no other decentring distortion methods would give large differences in behaviour either with the camera used. The leaning angle for the detector was calculated to 0.07 degrees with this camera.

6. Conclusions and future work

The general GCM model introduced in this chapter includes the best features from conventional camera models CCMs and models specialized for wide angle fisheye cameras in one unified model structure. Therefore there is no longer a need to use different models for these camera types. The GCM may also handle varying entrance pupil and decentring distortion as well as variations in focus and zoom.
In an experimental investigation it was concluded that the accuracy using the new radial distortion compensation in GCM performed better compared to the CCM model tested when few camera parameters were used. The additional distortions in GCM can be added for increased accuracy depending on the camera used. The decentring distortion is usually more important to include in models of cheap cameras, while "entrance pupil" variation can be important even for high precision cameras and mostly if it has a large viewing angle, and is more important for variable zoom and focus applications.
For future research we recommend to perform the accuracy investigation for different kinds of cameras, especially fisheye cameras. With a larger angle of view the advantage of the GCM should be bigger. These experiments should also be done with better leaning calibration images, so the compensation in Section 3.4 is not needed. Investigations on which kind of decentring distortion compensation is the best to use in practice should be done in such an investigation. It would be interesting to try the different "pre processing algorithms" e.g. with circular references, both for analysing how good parameter estimations can be achieved and also the accuracy of the "centre points" and calculation time. Another topic is to use natural corners in the environment as references. Also work on calculating analytical derivatives in the Jacobian for one or both of the transformation directions can be useful. To make calibrations with variable focus and zoom can also be of interest when the focus and zoom can be measured.

7. References

Bakstein, H. and T. Pajdla (2002). Panoramic mosaicing with a 180/spl deg/ field of view lens. *Proceedings of Third Workshop on Omnidirectional Vision, 2002.*

Basu, A. and S. Licardie (1995). Alternative models for fish-eye lenses. *Pattern Recognition Letters* 16(4): 433-441.

Brauer-Burchardt, C. and K. Voss (2001). A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images. *Proceedings of IEEE International Conference on Image Processing (ICIP)* Oct 7-10 2001, Thessaloniki, Institute of Electrical and Electronics Engineers Computer Society.

Brown, D. C. (1971). Close- range camera calibration. *Photogramm. Eng.* 37(8): 855-66.

Devernay, F. and O. Faugeras (2001). Straight lines have to be straight. *Machine Vision and Applications* 13(1): 14-24.

Dornaika, F. and R. Chung (2001). An algebraic approach to camera self-calibration. *Computer Vision and Image Understanding* 83(3): 195-215.

Fraser, C. S. (1997). Digital camera self-calibration. *ISPRS Journal of Photogrammetry and Remote Sensing* 52(4): 149-159.

Gennery, D. B. (2006). Generalized camera calibration including fish-eye lenses. *International Journal of Computer Vision* 68(3): 239-266.

Heikkila, J. (2000). Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22(10): 1066-1077. 0162-8828.

Kannala, J. and S. S. Brandt (2006). A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(8): 1335-1340.

Motta, J. M. S. T. and R. S. McMaster (2002). Experimental validation of a 3-D vision-based measurement system applied to robot calibration. *Revista Brasileira de Ciencias Mecanicas/Journal of the Brazilian Society of Mechanical Sciences* 24(3): 220-225. 0100-7386.

Nowakowski, A. and W. Skarbek (2007). Lens Radial Distortion Calibration Using Homography of Central Points. *Proceedings of EUROCON, 2007. The International Conference on "Computer as a Tool"*.

Olivier, D. F., L. Quang-Tuan, et al. (1992). Camera Self-Calibration: Theory and Experiments. *Proceedings of the Second European Conference on Computer Vision*, Springer-Verlag.

Pollefeys, M. and L. Van Gool (1999). Stratified self-calibration with the modulus constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21(8): 707-724.

Shah, S. and J. K. Aggarwal (1996). Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition* 29(11): 1775-1788.

Swaminathan, R. and S. K. Nayar (1999). Non-Metric Calibration of Wide-Angle Lenses and Polycameras. *Department of Computer Science, Columbia University, New York.*

Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Robotics and Automation.* 3(4): 323 - 344.

van Albada, G. D., J. M. Lagerberg, et al. (1995). A low-cost pose-measuring system for robot calibration. *Robotics and Autonomous Systems* 15(3): 207-227. 0921-8890.

Wei, G.-Q., K. Arbter, et al. (1998). Active self-calibration of robotic eyes and hand-eye relationships with model identification. *Robotics and Automation, IEEE Transactions on* 14(1): 158-166.

Zhou, F. and G. Zhang (2005). Complete calibration of a structured light stripe vision sensor through planar target of unknown orientations. *Image and Vision Computing* 23(1): 59-67.

Computer Vision
Edited by Xiong Zhihui

ISBN 978-953-7619-21-3
Hard cover, 538 pages
Publisher InTech
Published online 01, November, 2008
Published in print edition November, 2008

This book presents research trends on computer vision, especially on application of robotics, and on advanced approachs for computer vision (such as omnidirectional vision). Among them, research on RFID technology integrating stereo vision to localize an indoor mobile robot is included in this book. Besides, this book includes many research on omnidirectional vision, and the combination of omnidirectional vision with robotics. This book features representative work on the computer vision, and it puts more focus on robotics vision and omnidirectioal vision. The intended audience is anyone who wishes to become familiar with the latest research work on computer vision, especially its applications on robots. The contents of this book allow the reader to know more technical aspects and applications of computer vision. Researchers and instructors will benefit from this book.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen