Geometric Camera Calibration using Systems of Linear Equations*

Keith D. Gremban[†]
Martin Marietta Corporation

Charles E. Thorpe
Carnegie Mellon University

Takeo Kanade
Carnegie Mellon University

Abstract

Geometric camera calibration is the process of determining a mapping between points in world coordinates and the corresponding image locations of the points. In previous methods, calibration typically involved the iterative solution to a system of non-linear equations. We present a method for performing camera calibration that provides a complete, accurate solution, using only linear systems of equations. By using two calibration planes, a line-of-sight vector is defined for each pixel in the image. The effective focal point of a camera can be obtained by solving the system that defines the intersection point of the line-of-sight vectors. Once the focal point has been determined, a complete camera model can be obtained with a straightforward least squares procedure. This method of geometric camera calibration has the advantages of being accurate, efficient, and practical for a wide variety of applications.

1 Introduction

Many problems in computer vision and graphics require mapping points in space to corresponding points in an image. In computer graphics, for example, an object model is defined with respect to a world coordinate system. To generate an image, the points that lie on the visible surfaces of the object must be mapped onto the image plane; that is, 3d world points must be mapped onto 2d image points. In computer vision, the image locations of points on an object can be used to infer three-dimensional properties of the object; in this case, 2d image points must be mapped back onto the original 3d world points. In both cases, the mapping between 3d world coordinates and 2d image coordinates must be known. Geometric camera calibration is the process of determining the 2d–3d mapping between a camera and a world coordinate system.

We decompose the general problem of geometric camera calibration into two subproblems:

- The projection problem: given the location of a point in space, predict its location in the image; that is, *project* the point into the image.
- The back-projection problem: given a pixel in the image, compute the *line-of-sight* vector through the pixel; that is, *back-project* the pixel into the world.

A complete solution to the camera calibration problem entails deriving a model for the camera geometry that permits the solution of both the projection and the back-projection problems. For many applications, a complete solution is necessary. Some examples from the domain of mobile robots will help illustrate the problems.

*This material is based upon work supported by the Defense Advanced Research Projects Agency and the Army Engineering Topographic Laboratories under contracts DACA76-84-C-0005 and DACA76-85-C-0003, and by the National Science Foundation under grants DCR-8419990 and DCR-8604199. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency, the Army Engineering Topographic Laboratories, the National Science Foundation, or the U.S. Government.

[†]author's current address: Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213

In the CMU Navlab project [6], a robot vehicle follows roads using data from a color TV camera. In each image, the road is extracted, and the centerline and direction of the road computed in image coordinates. These parameters are then back-projected into vehicle coordinates and used to plot a course for the vehicle that stays within the road boundaries.

Turk, et al [8] describe a similar road following technique for the Autonomous Land Vehicle (ALV). Rather than parameterizing the road in terms of centerline and direction, they describe the road boundaries as a sequence of points. The line-of-sight vectors for each of the boundary points are computed by back-projection. The intersections of the line-of-sight vectors with the ground plane yield the points in the world between which the robot must steer to stay on the road. In addition, the ALV needs to know the predicted position of the road in each image. This prediction is obtained by projecting the location of the road into each image, based on the position of the road in the previous image, and the motion of the vehicle between images.

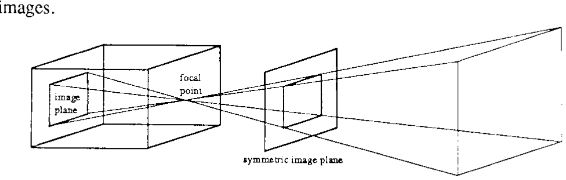

Figure 1: The Pinhole Camera Model

The simplest model for camera geometry is the pinhole, or perspective model. See figure 1. Light rays from in front of the camera converge at the pinhole and are projected onto the image plane at the back of the camera. To avoid dealing with a reversed image, the image plane is often considered to lie in front of the camera, at the same distance from the pinhole. The distance from the focal point to the image plane is the focal length.

A perfect lens can be modeled as a pinhole. No lens is perfect, of course, so part of the problem of geometric camera calibration is correcting for lens distortions. The most accurate and conceptually simple method of camera calibration would be to measure calibration parameters at each pixel in the image. For example, at each pixel measure the line-of-sight vector. This would produce a gigantic lookup table. Then, given a pixel in an image, simple indexing would yield the line-of-sight vector that solves the back-projection problem. To solve the projection problem, the table would be searched to find the line-of-sight vector that passes nearest the point in question.

A lookup table of calibration data for each pixel would be prohibitively expensive. The obvious compromise is to sample the image, and interpolate between data points. If the error in interpolation is less than the measurement error, no accuracy is lost. Most approaches to geometric camera calibration involve sampling the image, and solving for the parameters of the interpolation functions. The obvious differences between approaches are in the form of the interpolation functions, and the mathematical techniques used to solve for the parameters. The main intent of most calibration work has been the solution of the back-projection problem. The projection problem has occasionally been overlooked. The following paragraphs briefly describe past work.

- Sobel [5] introduced a method for calibration that involved the solution of a large system of non-linear equations. In addition to solving for the intrinsic camera parameters, his method also solved for extrinsic camera parameters, such as camera pan and tilt. Sobel used the basic pinhole model, and solved the system using a non-linear optimization method. He did not model lens distortions, and the system depended on the user to provide initial parameters for the optimization technique.

  Tsai [7] improved on the general non-linear approach in several ways. He modeled distortions globally using fourth order functions, and presented a method for computing good initial parameters for the optimization technique. Tsai's model of lens distortions assumes that the distortions are radially symmetric.

- Yakimovsky and Cunningham [9] presented a calibration technique that also used a pinhole model for the camera. They treated some combinations of parameters as single variables in order to formulate the problem as a system of linear equations. However, in this formulation, the variables are not completely linearly independent, yet are treated as such. No lens distortions are modeled with this approach.

- Martins, Birk, and Kelley [3] reported a calibration technique that does not utilize an explicit camera model. Their two-plane calibration method consisted of measuring the calibration data for various pixels across the image. The data for other pixels is computed by interpolation. The back-projection problem is solved by computing the vector that passes through the interpolated points on each calibration plane. The interpolation can be either local or global. The two-plane method solves only the back-projection problem.

  Isaguirre, Pu, and Summers [2] extended the two-plane method to include calibration as a function of the position and orientation of the camera. They used an iterative approach based on Kalman filters to obtain the solution.

Our goal in camera calibration was to develop a single, basic calibration procedure to solve both the projection and back-projection problems for a variety of applications. Consequently, the desired procedure had to be conceptually straightforward, easily extended to obtain various degrees of accuracy, and computationally efficient. To meet these requirements, we chose to begin with the two-plane method of Martins, Birk, and Kelley. Section 2 discusses the two-plane method and the solution to the back-projection problem. This method can be made arbitrarily accurate; the only problem is that it fails to solve the projection problem. In section 3 we present a method for solving the projection problem that utilizes the calibration data from the two-plane method. The solution to the projection problem is a simple application of analytic geometry, and is completely formulated with systems of linear equations.

The calibration method presented in this paper has been implemented and tested in the Calibrated Imaging Laboratory at CMU [4]. Results are presented in section 4 that demonstrate the accuracy of this method.

2 The Solution to the Back-Projection Problem

Martins, Birk, and Kelly [3] first formally presented the two-plane calibration technique for solving the back-projection problem. This technique has the advantage that it provides exactly the information needed—the ray in space that defines the line of sight of a given pixel—without any explicit camera model.

Figure 2 illustrates the concept of two-plane calibration. Let P_1 and P_2 denote the calibration planes. Assume that the 3d locations of the calibration points on each plane are measured. An image of each plane is acquired, and the image location of each of the calibration points is extracted. Let the calibration points be denoted p_{ij}, and the corresponding image locations be denoted q_{ij}, where $i = 1, 2$ is the plane, and $j = 1, 2, \ldots, n$ is the point index. Thus, the image of p_{ij} is q_{ij}.

Let ρ and γ denote the row and column coordinates, respectively, of an image. Then, given a point $v = [\ \rho \quad \gamma\ ]^t$ in the image, the line-of-sight vector for v can be computed as follows. First, use the points p_{1j} and q_{1j} to interpolate the location of v on the first calibration plane, P_1. Call this point u_1. Then, interpolate to find the location of v on the second calibration plane. Call this point u_2. The pixel line-of-sight vector then has direction $u_1 - u_2$ and passes through the point u_1.

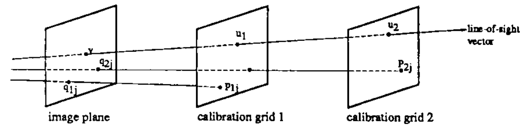

Figure 2: Two-Plane Calibration

Various types of interpolation can be used, with different degrees of accuracy. Martins, et al report three types of interpolation: linear, quadratic, and linear spline. The two-plane method has the potential for being the most accurate of any calibration method for the solution of the back-projection problem. At the limit, this technique consists of measuring the line-of-sight vectors for each pixel in the image. As will be seen in Section 4, the number of calibration points used has a strong influence on the accuracy of the calibration.

2.1 Global Interpolation

One approach to interpolating the calibration data is to globally fit an interpolation function to the data. This function is then used for any pixel across the entire image. Global interpolation has the effect of averaging errors over all the pixels so that the resultant line-of-sight vector is exact for no pixel, but is close for all pixels. This has the advantage of reducing the sensitivity to errors or noise in measurements. On the other hand, the form of the interpolation function is an *a priori* assumption about the lens distortions, and may or may not be appropriate.

2.1.1 Linear Interpolation

Let $p_{ij} = [\ x \quad y \quad z\ ]^t$, and $q_{ij} = [\ \rho \quad \gamma \quad 1\ ]^t$. Then a linear transformation between p and q is given by

$$p_{ij} = A_i q_{ij}$$

where A_i is a 3x3 matrix. Given n measurements on each plane, we can then form the system

$$[\ p_{i1} \quad p_{i2} \quad \ldots \quad p_{in}\ ] = A_i [\ q_{i1} \quad q_{i2} \quad \ldots \quad q_{in}\ ]$$

or,

$$P_i = A_i Q_i$$

This system can be solved in the least squares sense by using the matrix pseudoinverse (also called the generalized matrix inverse) [1]:

$$A_i = [Q_i^t Q_i]^{-1} Q_i^t P_i$$

Given a pixel v in the image, the direction of the line-of-sight vector through v is given by $u_1 - u_2$ where $u_1 = A_1 v$, and $u_2 = A_2 v$.

2.1.2 Quadratic Interpolation

Quadratic interpolation is similar to linear, except that second-order terms are used in the parameterization, and the matrix A is 6x6. We represent a point in space by $p = [\ x \quad y \quad z\ ]^t$, but we represent image locations by $q = [\ \rho \quad \gamma \quad \rho^2 \quad \gamma^2 \quad \rho\gamma \quad 1\ ]^t$. With these modifications, the formulation is otherwise identical. Martins, et al report that quadratic interpolation was more accurate than linear.

2.2 Local Interpolation

With no *a priori* knowledge about the lens distortions, global interpolation may be inappropriate. A better approach may be to model the distortions locally. If the calibration data is dense enough, the interpolation can be very accurate. In the paragraphs below, we discuss a technique called *linear spline interpolation*, which uses a linear function to perform interpolation over each local region.

Conceptually, this technique of interpolation consists of tesselating each calibration grid with triangles, and performing linear interpolation within each triangle. The calibration points form the vertices of the triangles. A plane is defined uniquely by three points, so no errors are introduced at the vertices. This is not the case for global interpolation techniques, in which errors are averaged over all points, including calibration points. Martins, et al achieved their best accuracy using this form of interpolation. In section 4, we report experiments which confirm this result.

In our current implementation, the grid is not tesselated in advance. Instead, for any point v in the image, each calibration grid is searched to find the three closest calibration points. The linear interpolation matrices A_i are computed using just three points each. The line-of-sight vector is then computed as in Section 2.1.1. Figure 3 illustrates the procedure.

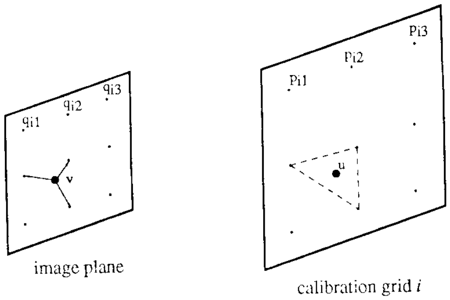

Figure 3: Linear Spline Interpolation

3 Linear Solution to the Projection Problem

The two-plane method for solution to the back-projection problem did not utilize an explicit camera model. In order to use the two-plane data to obtain a solution to the projection problem, it is necessary to have a camera model to formulate the equations. We use a model similar to that of Yakimovsky and Cunningham [9].

In figure 4 we begin with a pinhole model and define the following vectors and points:

- $P = [\ p_x \quad p_y \quad p_z\ ]^t$ == the vector from the origin to a point in space.
- $F = [\ f_x \quad f_y \quad f_z\ ]^t$ == the vector from the origin to the camera focal point.
- $R = [\ r_x \quad r_y \quad r_z\ ]^t$ == a vector that points along the direction of increasing row number. R represents the displacement vector from one pixel to the next in the row direction. The magnitude of R is the row scale factor.
- $C = [\ c_x \quad c_y \quad c_z\ ]^t$ == a vector that points along the direction of increasing column number. C represents the displacement vector from one pixel to the next in the column direction. The magnitude of C is the column scale factor.
- $[\ \rho_p \quad \gamma_p\ ]$ == the *piercing point* of the image, or the point where the optical axis pierces the image plane.

The vectors R and C define the orientation and scale of the image plane. Columns in the image plane are parallel to R, while rows are parallel to C.

The projection, $[\ \rho \quad \gamma\ ]$, of a point P onto the image plane can be computed by taking the dot product of the vector from the focal point to P, and adding the offset to the piercing point. For example, consider computing the row coordinate, ρ, of the projection. The vector $P - F$ is the vector from the focal point that passes through P. Every point along this vector will have the same location in the image. Let V be a normalized vector along $P - F$, that is, let

$$V = \frac{P - F}{\|P - F\|} \tag{1}$$

where $\|\cdot\|$ denotes the length of a vector. Let $\odot$ represent the usual vector dot product. Then $V \odot R$ represents the projection of V onto R measured in row units. Adding the row coordinate of the piercing point translates $V \odot R$ into image coordinates.

Therefore, the image location, $[\ \rho \quad \gamma\ ]$, of a point P can be computed using the equations:

$$\rho = V \odot R + \rho_p \tag{2}$$

$$\gamma = V \odot C + \gamma_p \tag{3}$$

If F, R, C, ρ_p, and γ_p are all unknown, then the resulting system is non-linear. However, the two-plane formulation of the back-projection problem yields the information needed to make solving for the focal point location a linear problem.

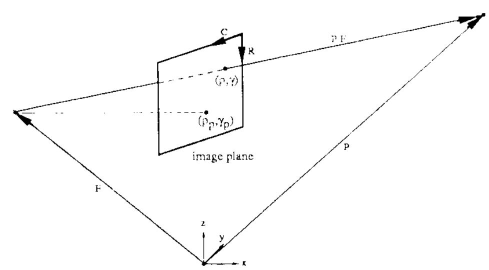

Figure 4: A Linear Model of Camera Geometry

3.1 Focal Point Solution

In a pinhole camera, all incoming light rays pass through the focal point. Since a lens is not a perfect pinhole, we instead refer to an *effective focal point*, which is the point that is closest to all the rays. From the two-plane method for the back-projection problem, one can compute a bundle of rays that pass through the lens. The next step is to find the point in space that minimizes the distance to all the rays.

The equation for the squared distance, d^2, from a point $P = [\ x \quad y \quad z\ ]^t$ to the line through $P_1 = [\ x_1 \quad y_1 \quad z_1\ ]^t$ in direction $[\ a \quad b \quad c\ ]^t$ (where $a^2 + b^2 + c^2 = 1$) is:

$$d^2 = \begin{vmatrix} y - y_1 & z - z_1 \\ b & c \end{vmatrix}^2 + \begin{vmatrix} z - z_1 & x - x_1 \\ c & a \end{vmatrix}^2 + \begin{vmatrix} x - x_1 & y - y_1 \\ a & b \end{vmatrix}^2$$

Expansion of terms yields:

$$\begin{aligned} d^2 &= x^2(b^2 + c^2) + y^2(a^2 + c^2) + z^2(a^2 + b^2) \\ &\quad - 2xyab - 2xzac - 2yzbc \\ &\quad + 2x(bk_3 - ck_2) + 2y(ck_1 - ak_3) + 2z(ak_2 - bk_1) \\ &\quad + k_1^2 + k_2^2 + k_3^2 \end{aligned}$$

where:

$$\begin{aligned} k_1 &= z_1 b - y_1 c \\ k_2 &= x_1 c - z_1 a \\ k_3 &= y_1 a - x_1 b \end{aligned}$$

To find the effective focal point, we need to minimize $D = \sum d^2$. Differentiating D with respect to x, y, and z yields:

$$\begin{aligned} \partial D/\partial x &= \sum 2x(b^2 + c^2) - \sum 2yab - \sum 2zac + \sum 2(bk_3 - ck_2) \\ \partial D/\partial y &= \sum 2y(a^2 + c^2) - \sum 2xab - \sum 2zbc + \sum 2(ck_1 - ak_3) \\ \partial D/\partial z &= \sum 2z(a^2 + b^2) - \sum 2xac - \sum 2ybc + \sum 2(ak_2 - bk_1) \end{aligned}$$

The sums are taken over all the line-of-sight vectors (a, b, c, k_1, k_2, k_3 are functions of the vectors).

Now, by setting the derivatives of D to zero to find the minima, and putting the equations in matrix form, we obtain:

$$h = Af$$

where:

$$h = \begin{bmatrix} \sum(ck_2 - bk_3) \\ \sum(ak_3 - ck_1) \\ \sum(bk_1 - ak_2) \end{bmatrix}$$

$$A = \begin{bmatrix} \sum(b^2 + c^2) & -\sum ab & -\sum ac \\ -\sum ab & \sum(a^2 + c^2) & -\sum bc \\ -\sum ac & -\sum bc & \sum(a^2 + b^2) \end{bmatrix}$$

$$f = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

So the solution we seek, f, the effective focal point of the camera, is simply:

$$f = A^{-1}h$$

3.2 Computation of the Camera Base Vectors

Equations (2) and (3) relate the position of a point in space to a corresponding image location. These equations can be written as a linear system:

$$\begin{bmatrix} \rho & \gamma \end{bmatrix} = \begin{bmatrix} v_x & v_y & v_z & 1.0 \end{bmatrix} \begin{bmatrix} r_x & c_x \\ r_y & c_y \\ r_z & c_z \\ \rho_p & \gamma_p \end{bmatrix}$$

Given N points in space, we have $2N$ equations to solve for the 8 unknowns in R, C, ρ_p, and γ_p:

$$\begin{bmatrix} \rho_1 & \gamma_1 \\ \rho_2 & \gamma_2 \\ \vdots & \vdots \\ \rho_n & \gamma_n \end{bmatrix} = \begin{bmatrix} v_{x1} & v_{y1} & v_{z1} & 1.0 \\ v_{x2} & v_{y2} & v_{z2} & 1.0 \\ \vdots & \vdots & \vdots & \vdots \\ v_{xn} & v_{yn} & v_{zn} & 1.0 \end{bmatrix} \begin{bmatrix} r_x & c_x \\ r_y & c_y \\ r_z & c_z \\ \rho_p & \gamma_p \end{bmatrix}$$

or,

$$B = WX$$

So, using the pseudoinverse to obtain a least squares solution, we have:

$$X = \left[W^t W\right]^{-1} WB$$

X contains the values for R, C, ρ_p, and γ_p.

3.3 The Local Projection Problem

In section 2 we presented several ways of modeling camera geometry for the back-projection problem. The *linear interpolation* technique (section 2.1.1), which involved fitting a first-order transformation to all the calibration data, is a global modeling technique. The *linear spline interpolation technique* (section 2.2) is a *local* modeling technique, since at each pixel, only the calibration data in a local region around the pixel is used to compute the interpolation function. The results of Martins, et al [3], and our laboratory results (see section 4) both indicate that a local modeling technique yields superior accuracy.

The solution to the projection problem presented in sections 3.1 and 3.2 is a global modeling technique. All the calibration data is used to compute the model parameters, and the results are used to solve the projection problem for any point in space. In direct analogy to the linear spline technique used in back-projection, a technique can be derived for using local information to improve the accuracy of solution to the projection problem.

Our local projection technique involves finding a linear model for local regions of the image. The technique involves two steps. In the first step, the global solution is used to obtain an estimated image location for a point in space. That estimated image location is used to find the four nearest calibration points on each calibration plane. These points are used to compute a local linear solution to the projection problem. The local linear solutions could also be precomputed, and the global solution would then be used simply to index the correct local solution.

4 Experimental Results

Measurements and tests were conducted within the Calibrated Imaging Laboratory (CIL) at CMU (Shafer [4]). The CIL is a facility that provides a precision imaging capability. The purpose of the CIL is to provide researchers with accurate knowledge about ground truth so that computer vision theories can be tested under controlled scientific conditions. Of particular interest for this study, the CIL provides facilities to accurately measure point locations, and to accurately position and orient cameras.

Position measurement of points in the CIL is performed with the use of theodolites (surveyor's transits), which are basically telescopes with crosshairs for sighting, mounted on accurate pan/tilt mechanisms. Objects to be measured are placed at one end of an optical bench; the theodolites are fixed to the other end, separated by a little more than 1 meter. To measure the position of a point, the crosshairs of each theodolite are placed over the point, and the horizontal and vertical displacements read off. Trigonometric equations then yield the position of the point in a Cartesian coordinate system defined with respect to the theodolites. As currently configured, the theodolites can determine point locations to less than 0.1 mm.

4.1 Test Scenario

The laboratory tests described below were designed to provide answers to the following questions:

1. What accuracies can be expected from off the shelf cameras and lenses?
2. How does increasing the number of calibration points affect the accuracy of calibration?
3. What is the expected accuracy for the projection problem?

Tests were performed using a calibrated grid. The grid consisted of horizontal and vertical lines 1mm in width, spaced 12.7 mm apart. The intersections of the lines on the grid were used as calibration points. A special intersection detector was implemented to extract the intersections from digital images with sub-pixel precision. Each time the grid was moved, new measurements were taken, an image digitized, and the intersection detector applied. The result was a data file in which each calibration point was associated with its 3d position and its image location.

A complete test consisted of data from three different grid locations. Due to the size of the laboratory, focal length of the lens, and depth of field of the lens, the grid was typically placed at distances ranging from 0.50m to 0.56m from the camera. Data from two of the grid locations was used to compute calibration parameters. These parameters were then tested using data from the third grid location. The third grid will often be referred to as the test grid. In each of the tests reported here, the focus of the camera was kept fixed. The camera used was a Sony CCD, model AVC-D1, with the standard 16mm lens.

A total of 300 calibration points were used on each grid. Rather than measure the location of each point individually, the location of each point was computed based on the measured locations of the center point and the four corners. Consequently, the accuracy of the data depended not only on the accuracy of the theodolites, but also upon factors such as the planarity of the grid, and the precision of the grid lines. In preparing for each test, the overall accuracy of the calibration data was estimated. For several points at each grid location, the 3d locations were measured using the theodolites. The measured locations were then compared with the computed locations. Differences of up to 0.2 mm were recorded, with typical differences being between 0.1 and 0.2 mm. The accuracy of the calibration method is limited by the accuracy of the calibration data, so the best accuracy achievable in this scenario is between 0.1 and 0.2 mm.

The effects of density of calibration points on calibration accuracy was tested by varying the number of calibration points used. This was easily implemented by simply skipping over some of the rows and columns in the grid. In each case, the calibration points were uniformly distributed over the image. Data is reported for the following distributions of points: 3x3, 5x7, 7x10, 15x20.

In all the tests reported below, grid 0 refers to the grid location farthest from the camera, while grid 2 refers to the grid location closest to the camera. In all cases, the number of points was varied to compute the calibration parameters, but all 300 calibration points on the test grid were used in testing.

4.2 Back-Projection Results

To test the accuracy of the back-projection problem, the image location of each of the calibration points on the third grid was used to compute a line-of-sight vector. The intersection of this vector with the plane of the test grid was computed, and the distance between the intersection and the actual position was used as the error measure. In the results reported below, the errors are averages taken over all the calibration points.

Table 1 presents the results obtained for the back-projection problem. The first error column contains the results for global linear interpolation. For this method, the density of the calibration grid makes little or no difference to the accuracy of the result. This was expected; since the calibration points are uniformly distributed across the grid, additional points do not provide additional information for a linear fit. The best accuracy is achieved when the test grid is positioned between the other two grids used for calibration.

array size	calibration grids	test grid	error (mm) global	error (mm) local
3x3	0, 1	2	1.921	0.731
	0, 2	1	0.388	0.296
	1, 2	0	0.561	0.317
5x7	0, 1	2	1.854	0.740
	0, 2	1	0.366	0.166
	1, 2	0	0.551	0.235
7x10	0, 1	2	1.810	0.696
	0, 2	1	0.350	0.169
	1, 2	0	0.509	0.185
15x20	0, 1	2	1.813	0.666
	0, 2	1	0.366	0.147
	1, 2	0	0.534	0.201

Table 1: Calibration Accuracy of the Back-Projection Problem

The second error column in table 1 presents the results obtained for back-projection problem using local linear spline interpolation. This time there is a general trend for greater accuracy with more calibration points. This reflects the fact that the linear spline method interpolates over local regions, and can more accurately approximate effects such as barrel distortion. There are instances observable in the table which seem to contradict the general trend; these are most likely due to noise in the measurements or in the process of point extraction. Over a number of trials, the general trend has been consistent.

The results in table 1 agree with the results obtained by Martins, et al. To summarize, the local linear spline interpolation procedure, with as few as 12 calibration points, is more accurate than global linear interpolation. In addition, the use of more calibration points improves the accuracy of the local linear spline method. The accuracies we achieved in our tests were at the level of the accuracies of our measurements.

4.2.1 Projection Results

The accuracy of the projection problem was tested with a procedure similar to that used in the back-projection problem. The 3d location of each calibration point on the test grid was projected into the image plane, and the difference (in pixels) between the projected location and the measured location was used as the error measure.

The test results for the projection problem are reported below in table 2. The first error column gives the error, in pixels, of the accuracy using global interpolation. The average error reported in all cases was less than two pixels, which is good enough for many applications. The results indicate that the standard lenses for our cameras are reasonably good, and can be approximated well with a pinhole model.

array size	calibration grids	test grid	error (pixels) global	error (pixels) local
3x3	0, 1	2	1.58	1.70*
	0, 2	1	0.89	0.65
	1, 2	0	0.89	0.82
5x7	0, 1	2	1.32	1.29
	0, 2	1	0.95	0.36
	1, 2	0	0.98	0.46
7x10	0, 1	2	1.20	1.01
	0, 2	1	0.91	0.34
	1, 2	0	0.88	0.40
15x20	0, 1	2	1.15	1.38*
	0, 2	1	0.92	0.92
	1, 2	0	0.91	0.36

Table 2: Accuracy of the Projection Problem

The second error column in table 2 reports the errors recorded using the local solution to the projection problem.

A comparison of the two columns in table 2 shows an improvement resulting from using local information. In general, the results from the local solution to the projection problem are a factor of 2 improved over the global solution. The entries followed by a * are examples where the global result was better than the local result; this may be an effect of errors in the measurement process. The general conclusion that can be drawn is that local models of camera geometry provide more accurate results than global models—for simple interpolation functions.

4.2.2 Conclusions

In section 4.1, we enumerated three questions which were to be answered by the tests reported above. We now proceed to answer each of these questions in turn.

1. *What accuracies can be expected from off the shelf cameras and lenses?*

   Tables 1 and 2 of test results show the accuracy achievable with a standard commercial CCD, using the standard lens supplied with the camera. With a simple global interpolation scheme, accuracies as good as 1 part in 1400 (0.3 mm over 530 mm) can be obtained. With a more sophisticated local linear spline interpolation, the accuracies can be increased to 1 part in 3500.

2. *How does increasing the number of calibration points affect the accuracy of calibration?*

   We have shown that the maximam accuracy for global linear interpolation can be achieved with a small number of calibration points, provided that the points are uniformly distributed over the image. Further increasing the number of calibration points has no effect on the accuracy. With a local linear spline interpolation, adding calibration points clearly improves the accuracy of the back-projection problem, until the limiting accuracy of the calibration data is reached.

3. *What is the expected accuracy for the projection problem?*

   Using either a local or global solution, the projection problem can be solved to within two pixels; results as good as 0.34 pixels were reported. For many applications, solution of the projection problem need not be extremely accurate. In many instances, the projected pixel location is only needed to find the center of a region within which an operation will be performed. For these applications, accuracy of one to two pixels is adequate.

It is important to note that the local interpolation outperformed the global interpolation. While the differences were not great in our tests, the lenses we used were fairly linear. If extremely wide angle lenses are used, the distortions may be large, and the ability to locally interpolate will be much more important.

5 Discussion

We have presented a calibration method that we believe meets many of the requirements of a basic calibration technique that can be used for a variety of applications. Our method is based on the two-plane method of Martins, Birk, and Kelley [3], but is extended to include a solution to the projection problem. We believe that the method presented here has many advantages, described in the following paragraphs:

- Completeness.

  The original two-plane method of calibration only provided a solution to the back-projection problem. While this is sufficient for many applications, a solution to the projection problem is also necessary for applications such as mobile robots. We have extended the two-plane method by providing a solution to the projection problem.

- Accuracy.

  The two-plane calibration method can be made arbitrarily accurate. As reported in section 4, increasing the number of calibration points results in increasing accuracy. If no improvement results from adding more points, then the accuracy of the calibration data must be improved.

- Simplicity.

  The two-plane model is conceptually very straightforward and easy to implement. The use of the line-of-sight vectors to solve for the parameters of a linear camera model arises intuitively from the geometry of the model. The method of solution for the camera model involves solving only linear equations, so no sophisticated optimization techniques are involved.

- Efficiency.

  Solution of either the back-projection or projection problems require only a few matrix multiplies and matrix inversions on small matrices. The operations are guaranteed to produce a unique answer within a fixed time. While a relatively large amount of data must be stored for this calibration method compared to other methods, the total amount is still insignificant.

- Practicality.

  Because the method provides a complete solution to the geometric calibration problem, the method can be used for any application. The accuracy can be arbitrarily increased (or decreased) to meet the requirements for a given application. The only change in the method is to store the data from more calibration points. The mathematics remains the same, and no special equipment is required beyond that needed to obtain precise locations for the calibration points.

In addition to the benefits of the method we presented, some general observations should be made:

- Without the benefit of *a priori* knowledge of the form of lens distortions, local modeling of distortions seems to perform better than global modeling. A global model is an attempt to fit the data into a predetermined form and average the error across the entire image. The accuracy of the interpolation is limited by how well the chosen model reflects reality. Conceivably, a different function could be required for different types of lenses to reflect different models of distortion.

  Modeling distortion locally makes no assumption about the forms of the lens distortions. The local model can be made arbitrarily accurate by simply sampling at more pixels. We have shown that relatively few points are needed to achieve the level of accuracy that the measurement devices provide. Moreover, local modeling is more accurate for solving both the projection and back-projection problems.

- Nearly all of the data reported showed that the best accuracy was obtained when the test grid was placed between the two grids used for calibration. This is a specific instance of the general fact that interpolation is more accurate than extrapolation. In calibrating a real robotic system, the calibration data should ideally be obtained so as to bound the region of interest as much as possible.

Geometric camera calibration may depend on a variety of factors. For example, the focal distance, the aperture setting, presence or absence of a filter, or even the operating temperature of a camera may all affect the calibration parameters. We are currently making measurements and conducting tests to determine the sensitivity of calibration parameters to many of these factors.

The results reported in this paper were obtained using data from the Calibrated Imaging Laboratory at CMU. Our next application of this method will be to calibrate and register three cameras and a laser range finder mounted on the CMU Navlab.

6 References

References

[1] A. Ben-Israel, and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, John Wiley & Sons, Inc., New York (1974)

[2] A. Isaguirre, P. Pu, and J. Summers, 'A New Development in Camera Calibration: Calibrating a Pair of Mobile Cameras,' *Proc. IEEE Conference on Robotics and Automation*, pp. 74-79 (1985).

[3] H. A. Martins, J. R. Birk, and R. B. Kelley, 'Camera Models Based on Data from Two Calibration Planes,' *Computer Graphics and Image Processing*, 17:173-180 (1981).

[4] S. A. Shafer, 'The Calibrated Imaging Lab Under Construction at CMU,' *Proc. 1985 DARPA Image Understanding Workshop.*

[5] I. Sobel, 'On Calibrating Computer Controlled Cameras for Perceiving 3D Scenes,' *Artificial Intelligence*, 5:185–198 (1974).

[6] C. E. Thorpe, M. Hebert, T. Kanade, and S. Shafer, 'Vision and Navigation for the Carnegie Mellon Navlab,' *Annual Review of Computer Science*, Volume 2, pp. 521–556 (1987), Annual Reviews, Inc.

[7] R. Y. Tsai, 'An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision,' *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 364–374 (1986).

[8] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, 'Video Road-Following for the Autonomous Land Vehicle,' *Proc. IEEE Conference on Robotics and Automation*, pp. 273–280 (1987).

[9] Y. Yakimovsky, and R. Cunningham, 'A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras,' *Computer Graphics and Image Processing* 7:195–210 (1978).