Complete Calibration of a Multi-camera Network

Patrick Baker and Yiannis Aloimonos
Computer Vision Laboratory
University of Maryland
College Park, MD 20742-3275

E-mail: {`pbaker, yiannis`}`cfar.umd.edu`

Abstract

We describe a calibration procedure for a multi-camera rig. Consider a large number of synchronized cameras arranged in some space, for example, on the walls of a room looking inwards. It is not necessary for all the cameras to have a common field of view, as long as every camera is connected to every other camera through common fields of view. Switching off the lights and waving a wand with an LED at the end of it, we can capture a very large set of point correspondences (corresponding points are captured at the same time stamp). The correspondences are then used in a large, nonlinear eigenvalue minimization routine whose basis is the epipolar constraint. The eigenvalue matrix encapsulates all points correspondences between every pair of cameras in a way that minimizing the smallest eigenvalue results in the projection matrices, to within a single perspective transformation. In a second step, given additional data from waving a rod with two LEDs (one at each end) the full projection matrices are calculated. The method is extremely accurate—the reprojections of the reconstructed points were within a pixel.

1. Introduction

In this paper, it is shown how to obtain the projection matrices for M cameras given N points, each projected into a subset of the M cameras. The calibration method uses every correspondence between every camera in a unified framework, based on an error measure which operates over all the camera correspondences at the same time. The calibration scheme is easy to implement for any number of cameras and requires no user intervention after the initial data collection. The past few years have seen some work in computer vision conducted in a multi-view environment (for example, work on image based rendering [12], multi-view structure from motion [11], multi-view human motion capture systems and multi-view virtualized reality [10]). In order for such techniques and the ones that will follow to produce accurate results, an accurate calibration is required.

The basic method calculates the projection matrices for all cameras to within a single perspective transformation. An extension to the scheme calculates the full projection matrices given additional input data, such as images of a line segment (rod) of known length, or the image of a calibration frame from two cameras.

This method is applicable to the calibration of large complexes of linear projective cameras of varying internal and external parameters. This method ensures that all the cameras are calibrated *as a single instrument* so that methods such as voxel carving and various algorithms in surveillance and monitoring can be implemented with maximal accuracy.

1.1. Contributions of the work and new ideas

The calibration procedure described here is a form of a multiframe structure from motion algorithm based on point correspondences, which are, by construction, of accuracy to within a pixel. The technique itself has a number of novelties as will become clear in the sequel, but there exist two innovations that make the procedure very robust and of general applicability. First, not all points need to be visible by all cameras simultaneously. This makes it possible to calibrate camera networks of unrestricted topologies, such as the following (Figures 1, 2 and 3). Figure 1 shows a map (blue print) of the corridors of a building floor with clusters of cameras placed at intervals along the walls. The technique described here can be used for calibrating this network. Figure 2a shows cameras arranged on the walls of a room and Figure 2b shows cameras arranged in a dome configuration. Our technique can be used here. Finally, Figure 3 shows a network of cameras looking outwards and arranged in some small volume of space. Our technique can be used

0-7695-0704-2/00 $10.00 © 2000 IEEE

here as well, but we would need to have the system in Figure 3 surrounded by other cameras, for example, by putting it in the room in Figure 2a or in the dome of Figure 2b. This way, we acquire information about the relative location of two cameras without a common field of view by relating them to surrounding cameras. The second innovation of the approach stems from a series of recent theoretical results in structure from motion. It has been shown [2, 3] that when a whole field of view (360 degrees) is available, the problem of recovering 3D motion and structure does not have the ambiguities and instabilities present in the case of a restricted field of view. Thus, in calibrating a multi-camera network we make sure that the collective field of view is such that it spans the whole space surrounding the area of interest. When, due to the topology of the network this is not possible, we use auxiliary cameras to achieve the full field effect. For example, consider that you wish to calibrate a stereo system. You could use three cameras (as in Figure 4) and perform the calibration procedure described here by waving a light in the space between the cameras. The outcome will be a robust calibration of the whole system and thus of the stereo system as well.

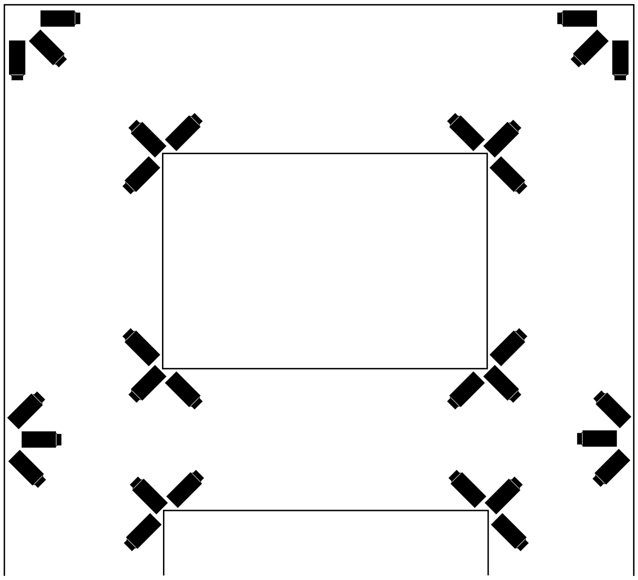
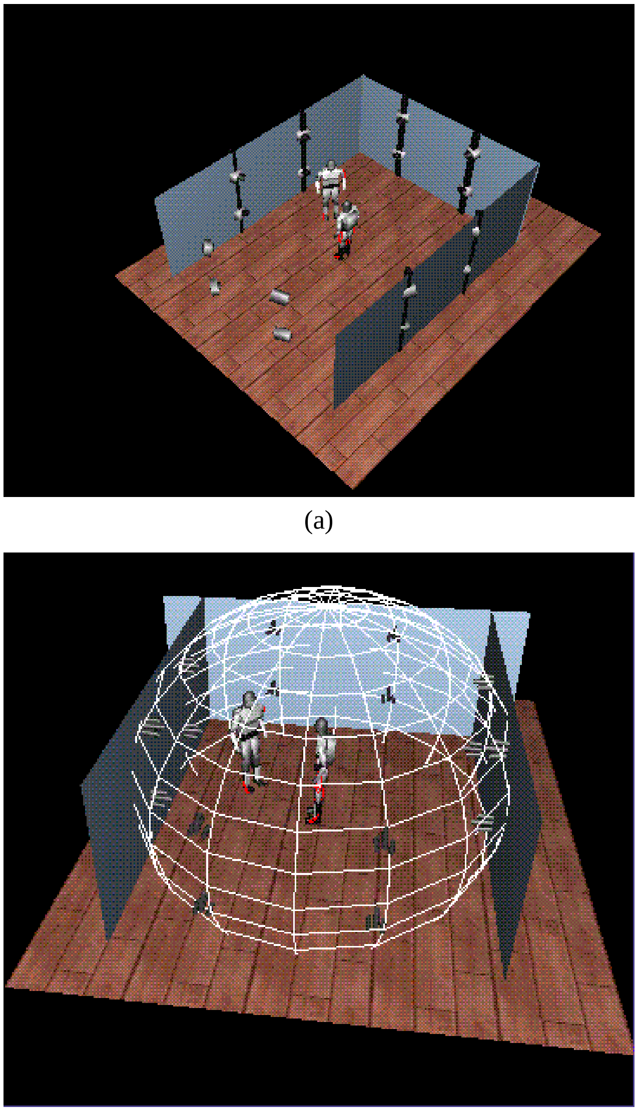


Figure 1. Camera network in a hallway.

1.2. Previous work

Previous calibration work has concentrated on the calibration of one camera with respect to a fixed reference calibration pattern, such as in [4]. One notable paper is [9] in which they ask if an accurate calibration pattern is really necessary. They answer their question in the negative through the use of at least three or more views from the same camera and simultaneous estimation of point location and calibration. They are also not concerned with external, but only internal calibration.

Calibration of three views has also been accomplished through the use of trilinear constraints [6, 13, 15] or quadrilinear constraints [6, 8, 15], but these tools are not well suited to the problem of calibrating many cameras at the same time. Spetsakis offers a solution in [14], but his algorithm uses only calibrated cameras, and additionally does not obtain the full projection matrices, even for calibrated cameras.



(a)

(b)

Figure 2. (a) Camera network in a room. (b) Dome-shaped camera network.

0-7695-0704-2/00 $10.00 © 2000 IEEE

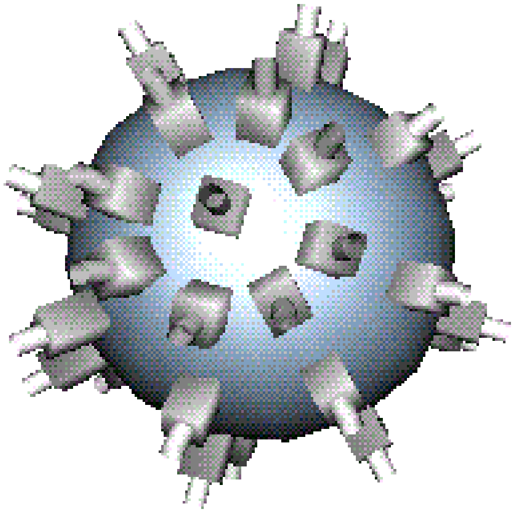


Figure 3. A compound-like eye composed of conventional video cameras.

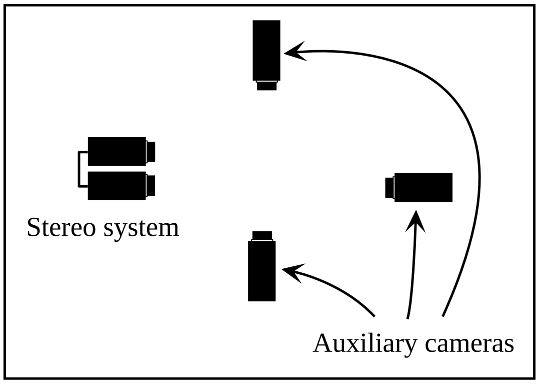


Figure 4. Scheme to calibrate a stereo system.

2. A multi-camera laboratory

2.1. Motivation

The motivation for the work in this paper came with the construction of several multi-camera laboratories all over the world. For a laboratory like that, see Figure 5. There are 64 cameras placed on the walls around the room. The cameras can be moved and refocused, so that calibration is needed with every data set. The cameras are synchronized, so that each frame in all cameras is taken at precisely the same time. It would be time consuming to locate the points of a calibration grid for each of the 64 cameras every time data is required. A calibration method with little to no user intervention is required.

2.2. Input data

The fact that the cameras are synchronized and that the lights can be turned off allows for many point correspondences. A calibrator waves around a rod with an LED on the end of it. The algorithm uses simple background subtraction and thresholding to extract the points. The correspondences are then used in a large nonlinear eigenvalue minimization routine, the basis of which is the epipolar constraint. The eigenvalue matrix encapsulates all the point correspondences between every pair of cameras in a way that minimizing the smallest eigenvalue results in the projection matrices.

3. Mathematical preliminaries

3.1. Notation

All vectors are column vectors, and transpose is used wherever necessary to keep the equations consistent. This is especially important in using the technique presented here, since many matrix morphology operators are used and a convention avoids confusion.

An $m \times n$ matrix A is represented as row vectors

$$A = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix}$$

where each $\mathbf{a}_i$ is an n vector. A may also be represented as

$$A = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,n} \\ \alpha_{2,1} & \alpha_{2,2} & \dots & \alpha_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m,1} & \alpha_{m,2} & \dots & \alpha_{m,n} \end{bmatrix}$$

3.2. Projective model

Using the standard projective model, a world point Q can be projected onto the image plane of an arbitrarily positioned unnormalized projective camera. Homogeneous coordinates represent $\mathbf{Q} = [X \ Y \ Z \ 1]^T$ and the projection can be modeled with a 3×4 matrix P which takes a homogeneous 3-vector $\mathbf{Q}$ and transforms it into a homogeneous 2-vector $P\mathbf{Q} = \mathbf{q} = s\,[x \ y \ 1]^T$.

P can be factored into the standard matrices K, R and T by using the QR decomposition to obtain $P = KR\,[I_3 \mid \mathbf{T}]$, where K is upper triangular, R is a rotation matrix, and $\mathbf{T}$ is a 3-vector. Note that contrary to convention, the camera is translated *first* then rotated. This is a necessary convention for this algorithm to work. These form the standard translation, rotation, and projection matrices. In the sequel, K and

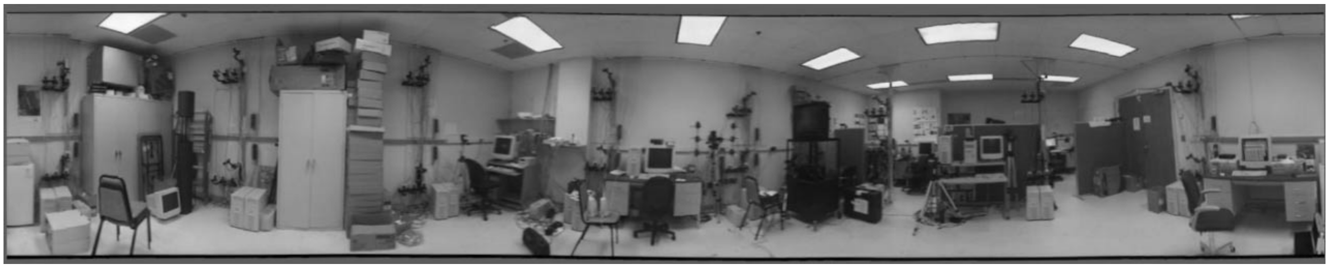


Figure 5. A panorama of a multi-camera laboratory

R separately will not concern us, and instead $B = KR$ is used.

Since the input to the algorithm is essentially a point cloud with no metric information, it is only possible to obtain calibration to within a 4×4 perspective projection H (see Theorem 1 in [5]), where

$$P_i H = P_i^*$$

Interestingly, ground control points are not needed in order find this H to within a conformal mapping. Only some sort of metric information must be specified. This metric information can be supplied by a wand with lights on both ends. The constant distance between the lights will calibrate the entire system to within a scale factor. If the distance between the lights is measured, then the scale factor is specified as well. This technique has the added benefit of being highly redundant so that distances are equalized throughout the entire camera field of view.

3.3. Morphology operators

Morphology operators which operate on 3×3 matrices are necessary for the derivations. For any 3×3 matrix M, the stacking operator is defined as:

$$[M]_s = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

The stack of a transpose of a matrix is as follows:

$$[M^T]_s = \begin{bmatrix} \mu_{1,1} \\ \mu_{1,2} \\ \mu_{1,3} \\ \mu_{2,1} \\ \mu_{2,2} \\ \mu_{2,3} \\ \mu_{3,1} \\ \mu_{3,2} \\ \mu_{3,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \mu_{1,1} \\ \mu_{2,1} \\ \mu_{3,1} \\ \mu_{1,2} \\ \mu_{2,2} \\ \mu_{3,2} \\ \mu_{1,3} \\ \mu_{2,3} \\ \mu_{3,3} \end{bmatrix} = D[M]_s$$

Two 3-vectors $\mathbf{u}$ and $\mathbf{v}$ can form $[\mathbf{u}\mathbf{v}^T]_s$. Setting $\mathbf{u}' = A\mathbf{u}$, where A is any 3×3 matrix, results in the following identity:

$$\begin{aligned}
[\mathbf{u}'\mathbf{v}^T]_s =& [A\mathbf{u}\mathbf{v}^T]_s \\
=& [A\mathbf{u}[v_1\ v_2\ v_3]]_s \\
=& \begin{bmatrix} A\mathbf{u}v_1 \\ A\mathbf{u}v_2 \\ A\mathbf{u}v_3 \end{bmatrix} \\
=& \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix} \times \begin{bmatrix} \mathbf{u}v_1 \\ \mathbf{u}v_2 \\ \mathbf{u}v_3 \end{bmatrix} \\
=& [A]_\alpha [\mathbf{u}\mathbf{v}^T]_s
\end{aligned}$$

where $[\cdot]_\alpha$ is defined here.

Deriving another identity with $\mathbf{v}' = A\mathbf{v}$ results in:

$$\begin{aligned}
\mathbf{u}\mathbf{v}'^T]_s =& D[\mathbf{v}'\mathbf{u}^T]_s \\
=& D[A]_\alpha [\mathbf{v}\mathbf{u}^T]_s \\
=& D[A]_\alpha D[\mathbf{u}\mathbf{v}^T]_s \\
=& [A]_\beta [\mathbf{u}\mathbf{v}^T]_s
\end{aligned}$$

where $[\cdot]_\beta = D[\cdot]_\alpha D$.

Given a vector $\mathbf{v}$, the skew symmetric matrix associated with it is defined as usual:

$$[\mathbf{v}]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

Finally, note that

$$[[\mathbf{v}]_\times]_s = K\mathbf{v}$$

where

$$K = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

These results will be used later in the paper.

4. Projective multiframe calibration

4.1. Problem statement

The camera system projects N 3D points, represented by $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_N$, onto M cameras, represented by projection matrices $P_1, P_2, \ldots, P_M$. A given point $\mathbf{R}_i$ is not necessarily projected onto camera j, but if it is, the projection is $\mathbf{r}_{i,j} = P_j \mathbf{R}_i$. If a particular point $\mathbf{R}_i$ does not project to a camera j, then $\mathbf{r}_{i,j} = [0\ 0\ 0]^T$ for notational convenience. The algorithm should output, given the points $\mathbf{r}$, the matrices P_j for all j. Since the camera system is calibrated using the LED method, it is assumed that there are many accurate point correspondences, so that minimality and robustness of data is not a concern. What is a concern is using all of the correspondences in an integrated minimization routine, so that all of the cameras are calibrated at one time.

4.2. Solution framework

Given the set of points $\mathbf{r}^*_{i,j}$ in cameras j, the algorithm searches for those matrices B_j^{-1} such that the $r_{i,j} = B_j^{-1} vecr^*_{i,j}$ are effectively formed in cameras which are only translations apart from each other. That is, all cameras have projection matrices of the form $P_j = [I_3 \mid \mathbf{T}_i]$. The algorithm measures this with the epipolar constraint, assuming that all the cameras are only translations apart from each other. Thus the algorithm can be seen as putting all the cameras into the translated copies of the same coordinate frame.

4.3. Normalized cameras

For the first derivation, it is assumed that all our cameras have projection matrices of the form $P_j = [I_3 \mid \mathbf{T}_i]$. This may seem to be an unnatural restriction, but the results will be extended later to general projection matrices.

Every pair of cameras j and k is constrained to have the sum of the squares of the epipolar constraint be zero. Simplified by the lack of rotation assumption, this is:

$$\sum_i (\mathbf{r}_{i,j}^T [\mathbf{T}_j - \mathbf{T}_k]_\times \mathbf{r}_{i,k})^2 = 0$$

Setting

$$A'_{j,k} = \sum_i [\mathbf{r}_{i,j}\mathbf{r}_{i,k}^T]_s [\mathbf{r}_{i,j}\mathbf{r}_{i,k}^T]_s^T$$

the above is equivalent to:

$$\begin{aligned}
[[\mathbf{T}_j - \mathbf{T}_k]_\times]_s^T A'_{j,k} [[\mathbf{T}_i - \mathbf{T}_k]_\times]_s &= 0 \\
(\mathbf{T}_j - \mathbf{T}_k)^T K^T A'_{j,k} K (\mathbf{T}_j - \mathbf{T}_k) &= 0 \\
(\mathbf{T}_j - \mathbf{T}_k)^T A_{j,k} (\mathbf{T}_j - \mathbf{T}_k) &= 0 \qquad (1)
\end{aligned}$$

where $A_{j,k} = K^T A'_{j,k} K$. Equation 1 just states that $\mathbf{T}_j - \mathbf{T}_k$ is an eigenvector of $A_{j,k}$ with zero eigenvalue.

This condition in equation 1 will not hold in the presence of noisy data, but the least eigenvector (eigenvector associated with the least eigenvalue) of $A_{j,k}$ should be the translation between cameras j and k.

This idea can be extended to be useful for many cameras. Shown next is a technique for forming a matrix whose least eigenvector $\mathbf{T}$ is the concatenation of all the camera positions $\mathbf{T} = [\mathbf{T}_1^T\ \mathbf{T}_2^T \ldots \mathbf{T}_{M-1}^T]^T$, where it is assumed without loss of generality that $\mathbf{T}_M = [0\ 0\ 0]^T$. To this end, form

$$A_i = \begin{bmatrix} A_{1,1+i} & & & & & \\ & A_{2,2+i} & & & & \\ & & \ddots & & & \\ & & & A_{M-i,M} & & \\ & & & & \ddots & \\ & & & & & A_{M,i} \end{bmatrix}$$

which is a block diagonal matrix containing the 3×3 epipolar constraints for all the cameras which are i indices apart, and zeros elsewhere.

Given the $3M \times 3M$ shift matrix

$$S_M = \begin{bmatrix} 0 & I_3 & 0 & \ldots & 0 \\ 0 & 0 & I_3 & \ldots & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & I_3 \\ I_3 & 0 & 0 & \ldots & 0 \end{bmatrix}$$

$L_{i,M}$ is defined to be the left $3M \times 3(M-1)$ submatrix of $I_{3M} - S_M^i$. $L_{i,M}$ is constructed so that T is in the nullspace of

$$C_i = L_i^T A_i L_i$$

The null space of $C = \sum_i C_i$ will be the intersection of the null spaces of the C_i. The only vector in this intersection is $\mathbf{T}$, if the cameras are in general position (no four coplanar). The constraint that C has one zero eigenvector is a constraint on all the correspondences among all cameras.

4.4. Calibrating the cameras

The method to calibrate the cameras is to find those matrices B_j, such that setting

$$\mathbf{r}_{i,j} = B_j^{-1}\mathbf{r}^*_{i,j}$$

where the star denotes the originally measured points, results in the C matrix constructed from the B_j having an eigenvalue close to zero. A nonlinear minimization over the B_j will minimize the smallest eigenvalue of C.

By using the previously derived morphological identities, it can be shown there is no need to multiply all the points to form $A'_{j,k}$ at every iteration. It is sufficient to multiply the $A'_{j,k}$ as follows.

$$\begin{aligned} A'_{j,k} &= \sum_i [r_{i,j} r_{i,k}^T]_s [r_{i,j} r_{i,k}^T]_s^T \\ &= \sum_i [B_j^{-1} r^*_{i,j} (B_k^{-1} r^*_{i,k})^T]_s [B_j^{-1} r^*_{i,j} (B_k^{-1} r^*_{i,k})^T]_s^T \\ &= [B_j^{-1}]_\alpha [B_k^{-1}]_\beta \left(\sum_i [r^*_{i,j} r^{*T}_{i,k}]_s [r^*_{i,j} r^{*T}_{i,k}]_s^T \right) [B_k^{-1}]_\beta^T [B_j^{-1}]_\alpha^T \\ &= [B_j^{-1}]_\alpha [B_k^{-1}]_\beta A'^*_{j,k} [B_k^{-1}]_\beta^T [B_j^{-1}]_\alpha^T \end{aligned}$$

thus obtaining

$$\begin{aligned} A_{j,k} &= K^T A'_{j,k} K \\ &= K^T [B_j^{-1}]_\alpha [B_k^{-1}]_\beta A'^*_{j,k} [B_k^{-1}]_\beta^T [B_j^{-1}]_\alpha^T K \end{aligned}$$

The minimization can compute the smallest eigenvalue of C over all of the B_i's using precomputed matrices $A'^*_{j,k}$.

4.5. Extracting the B matrices from the fundamental matrix

The nonlinear minimization must have initial conditions reasonably close to optimal. Since it is not guaranteed that correspondences exist between more than two cameras, the only possibility is to use the fundamental matrices to somehow extract an estimate of our B_i matrices. The `FMatrix` program by Zhengyou Zhang [16] was used to generate good fundamental matrices for input. Given that as input, the following generates the B_i matrices.

This factorization of the fundamental matrix is sufficient for our purposes. The fundamental matrix $F_{j,k}$ is defined by $\mathbf{r}_{i,k}^T F_{j,k} \mathbf{r}_{i,j} \approx 0$ for all i. This can be factored as:

$$F_{j,k} = B_k^{-T} [\mathbf{T}_k - \mathbf{T}_j]_\times B_j^{-1}$$

A single fundamental matrix is clearly not sufficient to extract any of the B_j's, but if it is only necessary to extract the B_j's to within a perspective transformation H, then all the B_j's can be obtained to within the *same* perspective transformation using only fundamental matrices.

Since the B's only to within a perspective projection are necessary, it is necessary to fix 12 parameters (3 more are fixed by setting $T_M = [0\ \ 0\ \ 0]$ and the last one is a scale factor). Nine are fixed by setting $B_1 = I_3$, and the three more parameters of B_2 are fixed as follows. The fact that $B_1 = I_3$ implies $F_{1,2} = B_2^{-1}[T_2 - T_1]_\times$. It is noted here that the SVD of $[T]_\times$ is

$$[T]_\times = \begin{bmatrix} -\mathbf{v}_2 & \mathbf{v}_1 & \mathbf{T} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1{}^T \\ \mathbf{v}_2{}^T \\ \mathbf{T}^T \end{bmatrix}$$

with $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{T}$ mutually orthogonal. Now $F_{1,2}$ can be decomposed using the SVD as follows,

$$\begin{aligned} F_{1,2} &= U \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1{}^T \\ \mathbf{v}_2{}^T \\ \mathbf{T}^T \end{bmatrix} \\ &= U \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} -\mathbf{v}_2{}^T \\ \mathbf{v}_1{}^T \\ \mathbf{T}^T \end{bmatrix} \begin{bmatrix} -\mathbf{v}_2 & \mathbf{v}_1 & \mathbf{T} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1{}^T \\ \mathbf{v}_2{}^T \\ \mathbf{T}^T \end{bmatrix} \\ &= B_2 [T]_\times \end{aligned}$$

with λ_3 arbitrary, and defining B_2. Thus setting λ_3 to a nonzero value will give us a nonsingular B_2 compatible with the perspective projection chosen by setting $B_1 = I_3$.

Now given B_1 and B_2 which fix the perspective projection, the rest of the B_i, consistent with this perspective projection, can be found given $F_{1,i}$ and $F_{2,i}$. Define

$$G'_i = F_{1,i} B_1 = B_i^{-T} [T_{1,i}]_\times \quad G''_i = F_{2,i} B_2 = B_i^{-T} [T_{2,i}]_\times$$

Taking the SVD of G'_i results in:

$$G'_i = U \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1{}^T \\ \mathbf{v}_2{}^T \\ \mathbf{T}^T \end{bmatrix}$$

This result can be expanded as follows:

$$\begin{aligned} & B_i^{-T} [T_{1,i}]_\times = \\ & U \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} \mathbf{v}_1{}^T \\ \mathbf{v}_2{}^T \\ 0 \end{bmatrix} + \begin{bmatrix} \alpha \mathbf{T}^T \\ \beta \mathbf{T}^T \\ \gamma \mathbf{T}^T \end{bmatrix} \right) \begin{bmatrix} -\mathbf{v}_2 & \mathbf{v}_1 & \mathbf{T} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1{}^T \\ \mathbf{v}_2{}^T \\ \mathbf{T}^T \end{bmatrix} \end{aligned}$$

Using the identity for $[T]_\times$ and taking the inverse transpose, results in:

$$B_i = U \begin{bmatrix} -\frac{1}{\lambda_1} & 0 & 0 \\ 0 & -\frac{1}{\lambda_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -v_2^T \\ v_1^T \\ w^T \end{bmatrix}$$

where w is an unknown vector. Requiring $B_i^T G''_i$ to be skew gives six linear conditions on w, which can be solved by least squares to get B_i. The result is the B_1, B_2, and B_i, which are all consistent with our three fundamental matrices. All the B_i can thus be used as initial conditions in the nonlinear minimization.

4.6. Obtaining the projection matrices

Note that the nonlinear minimization is actually carried out on the B_i. The T_i are extracted from the least eigenvector when the minimization is complete. The projection matrices are then completed as:

$$P_i = [B_i \mid B_i \cdot T_i]$$

4.7. Algorithm overview

Input.

M The number of cameras

N The number of points

$\hat{r}_{i,j}$ The corresponding points with $1 \leq i \leq M$ and $1 \leq j \leq N$.

Output. M projection matrices P_i

Steps.

- Find the initial matrices B_i using Section 4.5
- Using the B_i^{-1} as initial conditions, minimize the smallest eigenvalue of C to obtain the optimal B_i^* and C^*.
- From C^*, calculate the least eigenvector T^*.
- Using the B_i^* and T^*, calculate

$$P_i = [B_i^* \; B_i^* T_i^*] \qquad P_M = [B_M^* \; \mathbf{0}]$$

5. Extensions

5.1. Full multiframe calibration

The execution of the algorithm described above results in projection matrices P_i, which are related to the actual projection matrices P_i^* by $P_i^* = P_i H$, where H is a 4×4 matrix. To find H, a wand with lights on both ends is used.

Using the P_i, world points $R_{j,m}$ with $m \in \{1, 2\}$ can be reconstructed such that $R_j^* = H^{-1} R_{j,m}$. Since the wand does not change in length over the course of a data acquisition session, it is known that the distance between $R_{j,1}^*$ and $R_{j,2}^*$ is constant. Since the points are in homogeneous coordinates, one can formulate the condition on H^{-1} as follows.

$$\left| \frac{H_{1--3}^{-1} R_{j,1}}{H_4^{-1} Rj,1} - \frac{H_{1--3}^{-1} R_{j,2}}{H_4^{-1} Rj,2} \right|^2 = 1$$

where H_{1--3}^{-1} is the first three rows of H^{-1} while H_4^{-1} is the last row of H^{-1}. This condition can be used in a nonlinear optimization in order to find H.

5.2. Self calibration

To self-calibrate using this framework, it is assumed that there are correspondences from an uncalibrated camera with an unknown motion in a rigid environment. More precisely, this means that the B_i^{-1} can be decomposed as $R_i^T K^{-1}$, with K not dependent on i. The nonlinear minimization can then operate over the $3M + 5$ parameters defining the B_i^{-1} rather than the $8M$ parameters which define the B_i^{-1} up to scale.

One technical difficulty with this approach is that in order to use an unconstrained minimization, the R_i must be parameterized. Rodrigues parameters [1] are not appropriate, because these parameters are infinite for half rotations. More appropriate is Study's Soma [1], which is a homogeneous form of the Rodrigues parameters. If the Rodrigues parameters for a rotation R are $\mathbf{d} = [d_1 d_2 d_3]^T$, then the Soma parameters are $\mathrm{g} = s[1 \; \mathbf{d}^T]^T$, with s a scale so that g has length 1. Given unit g, [1] shows that R can be expressed as

$$R =$$

$$\begin{bmatrix} g_0^2 + g_1^2 - g_2^2 - g_3^2 & 2(g_1 g_2 - g_0 g_3) & 2(g_1 g_3 + g_0 g_2) \\ 2(g_2 g_1 + g_0 g_3) & g_0^2 - g_1^2 + g_2^2 - g_3^2 & 2(g_2 g_3 - g_0 g_1) \\ 2(g_3 g_1 - g_0 g_2) & 2(g_3 g_2 + g_0 g_1) & g_0^2 - g_1^2 - g_2^2 + g_3^2 \end{bmatrix}$$

6. Implementation details

This algorithm has been implemented in Matlab. Some implementation details which improve the performance of the algorithm follow.

6.1. Derivative of the C matrix

The B_j^{-1} which minimize the smallest eigenvalue of C are what is desired. A nonlinear optimization over the elements of B_j^{-1} for this smallest eigenvalue will obtain these parameters. To improve speed, it is possible to analytically take the derivative of the smallest eigenvalue of the C matrix with respect to each of the elements of the matrices B_j^{-1}. One just expands the sum which defines C and takes the derivative with respect to each matrix element in B_j^{-1}.

6.2. Preprocessing

It was shown in [7] that it is helpful in calibration to have reasonable normalization for the image points. This algorithm is no exception. While a projective calibration is obtained, the stability of the algorithm is improved if a coarse estimate K_e, j is made of the calibration matrix, for each $j \in [1, \ldots, M]$. Then the algorithm operates on the $\mathbf{r}'_{i,j}$ defined by

$$\mathbf{r}'_{i,j} = K^{-1} \mathbf{r}_{i,j}$$

6.3. Postprocessing

After the projection matrices are obtained, the calibration can be improved by reconstructing the world points, and then running a nonlinear optimization to obtain the projection matrices from the projections of the world points. The accuracy of the initial calibration, together with its self-consistency, allows the execution of this procedure.

7. Calibration results

The projective calibration algorithm has been implemented in Matlab, and has been used to calibrate a multi-camera configuration consisting of 64 cameras capable of synchronized recording. A typical data set contains approximately 2500 points, and each camera sees approximately half of those. It would be surprising if this method were not as accurate than a calibration frame, since it is possible to accurately locate corresponding points in every part of the cone of view of every camera. As long as the cones of view of the cameras intersect, then there should be sufficient data with which to calibrate.

In the experiments, it was determined that the reprojections of the reconstructed points were within a pixel, which was the measurement error of the location of the LED.

8. Conclusion

It has been shown that it is possible to calibrate a large system of cameras accurately with little user effort. This algorithm will be useful to calibrate a variety of multi-camera systems. In addition, as has been briefly explored here, the framework also allows for work on full Euclidean and self-calibration of a moving camera.

References

[1] O. Bottema and B. Roth. *Theoretical Kinematics*, chapter 6, pages 147–151. North-Holland Publishing Company, 1979.

[2] C. Fermüller and Y. Aloimonos. Geometry of eye design: Biology and technology. Technical Report CAR-TR-901, Center for Automation Research, University of Maryland, 1998.

[3] C. Fermüller and Y. Aloimonos. Observability of 3D motion. *International Journal of Computer Vision*, 1999. In press.

[4] A. S. for Photogrammetry". *Manual of Photogrammetry, 4th edition*. 1984.

[5] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 761–764. IEEE, 1992.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 1999.

[7] R. I. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997.

[8] A. Heyden. *Geometry and Algebra of Multiple Projective Transformations*. PhD thesis, Lund University, Sweden, December 1995.

[9] J.-M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration. In *5th European Conference on Computer Vision*, pages 158–174, 1998.

[10] P. Narayanan, P. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proc. International Conference on Computer Vision*, pages 3–10, Bombay, January 1998.

[11] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. International Conference on Computer Vision*, pages 90–95, 1998.

[12] S. M. Seitz and K. N. Kutulakos. Plenoptic image editing. In *Proc. International Conference on Computer Vision*, pages 17–24, 1998.

[13] M. Spetsakis and J. Aloimonos. A unified theory of structure from motion. In *DARPA IU Workshop*, pages 271–283, 1990.

[14] M. Spetsakis and Y. Aloimonos. A multi-frame approach to visual motion perception. *International Journal of Computer Vision*, pages 245–255, 1991.

[15] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*. Kluwer Academic Publishers, 1996.

[16] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, INRIA, Sophia-Antipolis, July 1996.