External camera calibration for synchronized multi-video systems

Ivo Ihrke Lukas Ahrenberg Marcus Magnor

Max-Planck-Institut für Informatik
D-66123 Saarbrücken

ihrke@mpi-sb.mpg.de ahrenberg@mpi-sb.mpg.de magnor@mpi-sb.mpg.de

ABSTRACT

We present a method for external camera calibration that is simple to use and offers generality in the positioning of the cameras. This makes it very suitable for the calibration of mobile, synchronized camera setups. We use a *camera graph* to perform global registration which helps lifting restrictions on the camera setup imposed by other calibration methods. A further advantage is that all information is taken into account simultaneously. The method is based on a virtual calibration object which is constructed over time by tracking an easily identifiable object through three-dimensional space. This implies that no calibration object must be visible simultaneously in all cameras.

Keywords

External Camera Calibration, Registration, Multi-Video System, Graphs

1 Introduction

We have built a mobile camera system to take computer vision research to more general settings than is possible with a fixed studio setup. This added flexibility demands a simple to use camera calibration technique that allows for convenient calibration of the camera system.

Since camera calibration is such a crucial task in computer vision, a lot of effort has been spent on the subject. There exist a number of methods in different flavors [Tsa87; HZ00]. Most of them use carefully prepared calibration objects, while others estimate camera parameters from general images or image sequences. These approaches are called self-calibrating. A relatively new approach is the use of a virtual calibration object. This calibration object does not exist in a physical meaning, but is instead constructed over time (assuming a static camera setup and scene) by tracking an easily identifiable object through the 3D scene. The point correspondences hereby obtained are then used for calibration [AP95; CDS00].

The advantages of using virtual calibration objects are the simple establishment of point correspondences in difficult, wide baseline and encircling camera setups as well as the possibility of acquiring as many point measurements as necessary, which is difficult in one image approaches. Furthermore, there is no need for a carefully manufactured calibration object, which can break during transportation.

The paper is organized as follows. In Section 2 we discuss some previous work that is important in this context. Section 3 gives an outline of the method, Section 4 describes the establishment

Journal of WSCG, Vol.12, No.1-3., ISSN 1213-6972

WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.

of 2D correspondences for calibration purposes. In Section 5 the computation of pairwise relationships between cameras is discussed. Section 6 describes the main contribution of this paper, namely the usage of a graph structure for external camera calibration. Section 7 presents experiments with synthetic as well as with real data. Finally, Section 8 concludes the paper and gives some directions for future work.

2 Related Work

Earlier approaches to external calibration with help of virtual calibration objects [AP95; CDS00] use light emitting objects (i.e. flashlight, LED) in dark rooms for tracking. In [AP95] the virtual calibration object is introduced to calibrate a pair of stereo cameras. A synchronized pair of cameras is used to track a flashlight, the path of which is used for calibration. Chen et. al. extend this work by applying the virtual calibration object approach to unsynchronized multi-camera setups. The lack of camera synchronization is treated with an Extended Kalman Filter (EKF) which estimates the path of a marker object (LED). Pairwise relationships between cameras are computed employing a structure-from-motion algorithm. Global registration in a common coordinate system is then done using a triangulation scheme iteratively, to estimate the position of yet unregistered cameras to two already registered cameras [CDS00]. This requires that any unregistered camera is connected to at least two already registered cameras by known pairwise relationships.

Our approach removes this restriction by analyzing a graph consisting of the cameras as vertices and known pairwise position and orientation as graph edges. An additional advantage is that all information about the global position of a given camera is imposed simultaneously.

3 Graph-based external multi-camera calibration

To accommodate general camera setups, we use an approach that does not require visibility of the calibration object from all cameras. Instead, a *virtual* calibration object that covers the working volume is constructed over time by tracking an easily identifiable object [AP95; CDS00]. We mainly follow the approach of [CDS00]. However, we lift the constraints that the cameras must be registered by triangulation from the base camera pair and that the working volume must be dark during calibration.

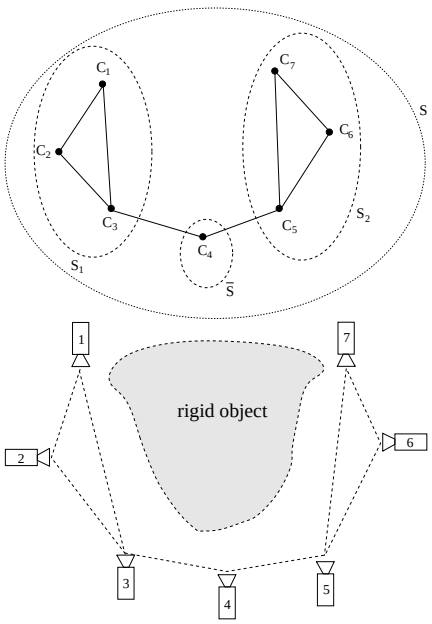

Figure 1: An example for a graph with nonempty sets $S, S_k, \bar{S}$ (upper image) and a corresponding scene where this situation could occur (lower image).

Intrinsic calibration is done independently for each camera using Tsai's method [Tsa87]. This is reasonable since our cameras do not allow for varying internal parameters. This requires us to record an image of a checkerboard for each camera, from which the internal parameters can be computed. However, the checkerboard does not have to be visible from all cameras simultaneously.

At the place of recording, we first obtain a number of 2D correspondences via tracking the marker object depicted in Fig.2. The midpoints of the marker object are determined and the correspondences are used to robustly compute the relative orientation of each camera pair [Zha96; Hor90].

Since the pairwise relative position of the cameras can be found up to a scale factor only, a global registration has to be performed to achieve global calibration. This is done by constructing a graph $G = (V, E)$ that represents the relationship between cameras, Fig. 1. The cameras $C_i \in V$ are the vertices, and known relative position and orientation between any pair of cameras $(C_i, C_j) \in E$ are represented by the edges of the graph. The graph is undirected, since known relative position and orientation $(\mathbf{R}, \mathbf{t})$ for a camera pair (C_i, C_j) also defines the relative position and orientation of (C_j, C_i) as $(\mathbf{R^T}, -\mathbf{R^T t})$.

Once the graph is set up, it is searched for cyclically connected subsets $S_k \subseteq S$ containing all

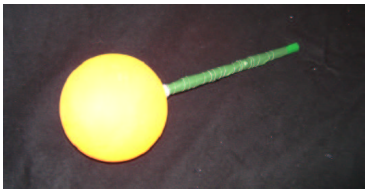

Figure 2: The marker object that is tracked through the scene to establish 2D correspondences between camera frames consists of a bright orange ball on a stick.

cameras. The set $\bar{S} := S \setminus (\cup_k S_k)$ consists of all cameras not belonging to any cycle. If $S = \bar{S}$, no cycles exist and a special treatment is necessary. Figure 1 illustrates a situation where all sets are nonempty.

The unknown scale factors for the pairwise translations are determined for each S_k independently. This is done by solving an over-determined linear system of equations. The equations correspond to cycles in the graph and require that the scaled translations along any closed path sum to zero. That means the cycle represents a closed curve in three-dimensional space. Since the estimation obtained this way is not connected to the image measurements any longer, the reprojection error of this solution may not be optimal yet. Nevertheless, the overall shape of the setup is recovered quite well. Therefore, this estimate is used as an initial estimate for bundle adjustment [TMHF00; HZ00]. Using these partial registrations, parts of the virtual calibration object (i.e. the path of the tracked marker object) are reconstructed. The remaining step is to register all subsets S_k and all $C_i \in \bar{S}$ with each other to form a globally consistent calibration.

In this framework the improvement of our method can be stated like this: The restriction of the vertices V being interconnected by three-cycles in the graph G as required by the algorithm of Chen et. al. [CDS00] is lifted, and arbitrary connectivity of the graph G is allowed for as long as the graph is not disconnected.

4 Obtaining image correspondences

2D point correspondences between images, i.e. projections of the same three-dimensional point onto different camera planes, are the basis for epipolar geometry estimation and therefore are the first step in the recovery of the camera structure.

We obtain them by tracking a sphere over time, establishing one correspondence for every frame of the calibration video sequence. The sphere is a suitable object because it has a unique midpoint, the projections of which can be computed from images alone. The sphere projects to a conic section in the image plane as shown in Fig. 3. In any real situation this conic will be an ellipse. If the focal length of the cameras is not too small the ellipse is nearly circular. This observation leads to our sphere detection algorithm.

The sphere detection is run on every frame of the video sequences separately. Our marker object has a color which is not widely present in the scene. Therefore we threshold the image with a color band in Y/Cb/Cr space. We find connected components in the resulting binary image and threshold them according to their size to exclude noise and small artifacts from further processing. A circular Hough transform is performed on every remaining connected component. This yields a best circular fit for each of them.

To find the best fit for the image, the Hough scores can not be compared directly as they depend on the size of the object. Therefore, we calculate the density of object pixels in the circle candidates. A candidate from a round component will typically fill the whole disk and gain a high score. The highest scoring circle is picked as the winner. Finally, the score is thresholded. If the score is higher or equal to the threshold we have found the sphere, otherwise we assume that the calibration object is not visible in the image.

If the sphere was found, we refine the estimate using orthogonal least squares ellipse fitting. For this purpose we extract the edges of the original image around the position found by the Hough transform. The area for edge detection is limited by the estimated radius plus some safety region. The detected edge pixels are used to robustly fit an ellipse using the RANSAC paradigm [FB81]. We use the squared sum of orthogonal euclidean distances between the data points and the ellipse as an error measure. This gives a sub-pixel estimate of the ellipse midpoint and compensates for small mislocalizations introduced by the Hough transform. We use midpoints in corresponding frames of the multi-video sequence as correspondences.

5 Computation of pairwise position and orientation

Having extracted a set of point correspondences, we can proceed in computing the pairwise relative position and orientation $(\mathbf{R_{ij}}, \mathbf{t_{ij}})$ of any camera pair (C_i, C_j).

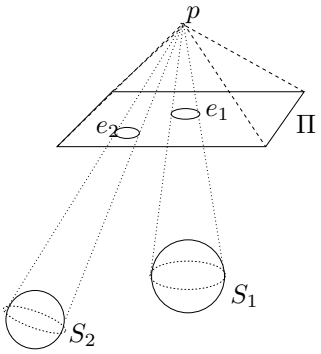

Figure 3: Projection of spheres S_1 and S_2 onto the image plane Π using the focal point p. S_1 projects to a circle, because the main axis of its cone of projection is perpendicular to Π. S_2 on the other hand projects to an ellipse.

This is achieved by first undoing the effects of the internal camera parameters on the midpoints, followed by the computation of the essential matrix $\mathbf{E_{ij}}$. We use a variant of Zhang's approach [Zha96] to compute the essential matrix. The difference between Zhang's algorithm and our version is the initial guess of the fundamental matrix $\mathbf{F_{ij}}$, which we obtain using RANSAC [FB81] with the seven point algorithm for estimation of the fundamental matrix [HZ00]. This estimate already fulfills the rank two constraint on the fundamental matrix $\mathbf{F_{ij}}$. The following nonlinear refinement step minimizes the symmetric epipolar distance, i.e. the distance of the epipolar line induced by a point to the corresponding point in the other image, using the Nelder-Mead simplex method [NM65]. Given the internal parameter matrices $\mathbf{K_i}$ and $\mathbf{K_j}$, an initial guess of the essential matrix is computed $\mathbf{E_{ij}} = \mathbf{K_j}^T \mathbf{F_{ij}} \mathbf{K_i}$. This guess usually does not fulfill the additional constraint that the two nonzero singular values of $\mathbf{E_{ij}}$ are equal. The nearest essential matrix fulfilling the constraints is obtained by setting the two unequal singular values to their mean value [FL01]. However if the two nonzero singular values differ widely, this is not a good guess for the essential matrix minimizing the symmetric epipolar distance. Therefore, an additional nonlinear minimization with the simplex-algorithm is performed, this time using only the five parameters of rotation and translation direction to compute the final guess for matrix $\mathbf{E_{ij}}$.

The essential matrix is of the form $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$, and rotation as well as translation can be extracted from it [Hor90]. The decomposition of matrix $\mathbf{E}$ yields four solutions for $\mathbf{R}$ and $\mathbf{t}$. The solutions arise because of the unknown scale of $\mathbf{t}$ which can be either positive or negative. The rotations are related by a rotation of 180° around the baseline, connecting the two cameras. In most cases, this ambiguity can be resolved by the demand that reconstructed points are in front of both cameras [HZ00; FL01]. Sometimes, in cases of a near-180° angle between the principal axes of the two cameras, it is only possible to resolve a twofold ambiguity and two solutions remain. Since the reprojection error of both solutions is the same, there is no measure based on image distances that can be chosen to decide the solution. Therefore, we apply a heuristical measure: we choose the solution whose reconstruction has a smaller convex hull. Since we are dealing with euclidean reconstructions this is a reasonable choice.

As a measure of quality, we compute a local reconstruction for each camera pair (C_i, C_j) that yields a solution $(\mathbf{R_{ij}}, \mathbf{t_{ij}})$ and evaluate the reprojection error caused by that reconstruction. We apply a threshold on that error to exclude unstable estimates.

6 Graph building and analysis

From the previously computed pairwise positions and orientations, we build a graph that mirrors the availability of rotation/translation information for all camera pairs. The vertices of the graph correspond to the cameras, and edges connecting them indicate a stable solution to the relative position and orientation problem for the cameras connected by that edge. Looking at the problem from a graph theoretical point of view has several advantages. There are standard solutions for problems occurring in general camera setups. It is for example simple to check if a calibration containing all cameras is possible. It is sufficient to check that the graph is connected. This can already be done during data collection and the user can be guided to create more correspondences for camera pairs that have not sufficient data available yet.

If there are unconnected subsets of vertices in the graph, no global calibration can be found, but the subsets of cameras can be calibrated separately. Cameras corresponding to isolated vertices cannot be calibrated, because they do not share any overlapping viewing space with the other cameras.

6.1 Registration of cyclically connected components

We proceed by registering all cameras C_i^k contained in a cyclically connected component G_k of

the graph G with each other. This is done for all k separately. For improved readability we skip the subscript k in the further discussion.

Recall that all relative translations $\mathbf{t_{ij}}$ are defined only up to a scale factor and reside in the local coordinate system of camera C_i. The task of the registration procedure is to find consistent scale factors s_{ij} for the $\mathbf{t_{ij}}$ and their transformations to a global coordinate system shared by all cameras.

The registration is based on cycles in the graph $G = (V, E)$. We denote a cycle as $Z = [z_1 \dots z_n], z_i \in \{C_1 \dots C_m\}$ and introduce $\tilde{\mathbf{t}}_{\mathbf{ij}}$ as translation direction $\mathbf{t_{ij}}$ transformed to a common coordinate system. Then one condition on the scale factors s_{ij} for one cycle can be written as:

$$\left(\sum_{i=1}^{n-1} s_{z_i,z_{i+1}} \tilde{\mathbf{t}}_{z_i,z_{i+1}} \right) - s_{z_n,z_1} \tilde{\mathbf{t}}_{z_n,z_1} = 0 \qquad (1)$$

(1) is valid for all cycles $Z^h, h \in \{1 \dots l\}$. It describes the condition that every walk along a cycle in the registered set of cameras returns to its origin, or all translations along a cycle sum up to zero. Every cycle contributes three equations to the system of equations in (1), one for each coordinate. Overall, there are $3l$ equations that make up a usually overdetermined linear system of equations $\tilde{\mathbf{T}}\mathbf{s} = \mathbf{0}$, which can be solved to obtain the scale factors s_{ij}. Matrix $\tilde{\mathbf{T}}$ and vector $\mathbf{s}$ contain the t_{ij} and s_{ij} in appropriate positions, respectively.

To apply this to the registration, a number of preliminary steps are necessary which will be explained in more detail later:

1. find all cycles Z^h in graph $G = (V, E)$,
2. transform all $\mathbf{t_{ij}}$ into a common coordinate system, i.e. compute $\tilde{\mathbf{t}}_{\mathbf{ij}}$,
3. construct and solve the system of linear equations in equation (1),
4. use the computed scale factors s_{ij} to compute a consistent calibration

1) An algorithm for computing all cycles in a graph is presented in [KYD+03].

2) Without loss of generality, we choose one of the cameras local coordinate systems as the common coordinate system. The corresponding camera is referred to as base camera C_b. We use the camera related to the vertex with the highest degree for that purpose. This choice minimizes the number of edges that have to be traversed to transform the other cameras into the base camera's coordinate system.

For a given camera C_i we transform the translation directions $\mathbf{t_{ij}}, (C_i, C_j) \in E$ originating in C_i into the common coordinate system, yielding $\tilde{\mathbf{t}}_{\mathbf{ij}}$. This is done using the quaternion weighted mean rotation $\overline{\mathbf{R}}_{ib}$ between C_i and C_b: Rotation $\overline{\mathbf{R}}_{ib}$ is computed by finding all paths between cameras C_i and C_b, accumulating rotation matrices along these paths and taking the weighted mean value of their quaternion representations using iterative spherical linear quaternion interpolation[1].

3) The solution $\mathbf{s}$ of the linear system $\tilde{\mathbf{T}}\mathbf{s} = \mathbf{0}$ is defined up to a scale factor only. Therefore, matrix $\tilde{\mathbf{T}}$ is singular. This is because the overall scale factor of the camera setup cannot be determined from image measurements alone. To solve the linear system, we compute the basis vector of the right null-space of $\tilde{\mathbf{T}}$. Using the standard approach to this problem, we compute the singular value decomposition (SVD) of matrix $\tilde{\mathbf{T}}$: $\tilde{\mathbf{T}} = \mathbf{UDV^T}$ and extract the column of $\mathbf{V}$ corresponding to the zero singular value.

4) To compute a consistent calibration, we multiply the transformed translations $\mathbf{t_{ij}}$ by their corresponding scale factors s_{ij}. If the solution was perfect, the calibration would be readily available since every path from C_i to C_b in the graph could be taken to transform camera C_i into the base coordinate system. But because $\mathbf{s}$ is a least-squares solution this is generally not true. To get the best estimate of the true translations, we compute the weighted mean translation $\overline{\mathbf{t}}_{\mathbf{ib}}$ similarly to the computation of $\overline{\mathbf{R}}_{\mathbf{ib}}$ in step 2). The calibrated cameras $\overline{C}_i$ are now defined as having the projection matrices

$$\mathbf{P_i} = \mathbf{K_i}\overline{\mathbf{R}}_{\mathbf{ib}} \begin{pmatrix} \mathbf{1} & | \overline{\mathbf{t}}_{\mathbf{ib}} \end{pmatrix} \qquad (2)$$

with $\overline{\mathbf{R}}_{\mathbf{bb}}$ being the unit matrix and $\overline{\mathbf{t}}_{\mathbf{bb}}$ the zero translation for the base camera C_b.

6.2 Bundle adjustment

The calibration obtained so far captures the coarse structure of the camera setup quite well, but since it minimizes an algebraic error, namely the least-squares error of equation (1) which has no physical meaning, the reprojection error of the points reconstructed with this calibration is not optimal yet. Nevertheless, it serves as a good initial guess for bundle adjustment [TMHF00]. We implemented a reduced version which optimizes

[1] The weighted mean value $\overline{x} = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i x_i$ can be computed in an iterative manner, where a linear interpolation is performed in every iteration. Replacing the x_i by quaternions and using spherical linear interpolation a weighted mean value of rotations can be computed.

standard deviation of uniform Gaussian noise in pixels	mean reprojection error left image in pixels	mean reprojection error right image in pixels
0.1	0.0213	0.0200
0.3	0.0710	0.0677
0.5	0.1174	0.1131
0.7	0.1600	0.1512
0.9	0.2045	0.2151
1.1	0.2548	0.2425
1.3	0.2815	0.2718
1.5	0.3492	0.3262
1.7	0.3612	0.3452
1.9	0.4503	0.4263

Table 1: Mean reprojection error in the left and right images for pairwise relationship computation (inliers only) for different noise levels (experiment 1).

standard deviation of uniform Gaussian noise	mean reprojection error over all camera pairs	mean reprojection error after bundle adjustment
0.1	0.1474	0.1003
0.3	0.7382	0.3285
0.5	1.4748	0.5481
0.7	1.2274	0.7602
0.9	1.5640	1.0145
1.1	1.8092	1.2005
1.3	2.0367	1.4523
1.5	2.7336	1.6161
1.7	8.6253	2.1588
1.9	6.0498	2.1608

Table 2: Mean reprojection error of the reconstruction from the linear solution to the calibration dependent on the noise level and after bundle adjustment (experiment 2).

only rotation and translation parameters of the final calibration using a nonlinear optimization method. Usually, all parameters (internal, external and reconstructed point positions) are optimized which is a huge problem with hundreds of variables and can only be handled by exploiting the coarse structure of the problem [Pol99].

We parametrize the problem in a similar way as in the essential matrix refinement step, but this time for all cameras simultaneously. The rotation is parametrized as the normalized rotation axis scaled by the rotation angle. These values are obtained from the quaternion representation of the rotation. The translation values are used directly for the parametrization. As the error measure that is to be minimized, we adopt the sum of the mean reprojection errors of the reconstructed points in all cameras. This error measure is optimized using the simplex algorithm and the previous calibration as an initial guess.

7 Experiments

We performed three experiments to validate our calibration method. The first two experiments use synthetic data with different levels of additive uniform Gaussian noise to assess the robustness of our method to noise. The first experiment shows the accuracy of the pairwise position and orientation computation (see Section 5) under noise. The second experiment evaluates the same for the initial calibration obtained by solving the linear system of scale factors, Section 6.1 and the improvements of bundle adjustment, Section 6.2. The third experiment uses real data and is the calibration of our in-house multi-video studio [TLMS03], which is obtained using the full method. Errors are mean reprojection errors in pixels if not indicated otherwise.

7.1 Performance of the pairwise relationship estimation

For this experiment we use 100 synthetic 3D data points. The data points are projected to 5 cameras and disturbed by uniform Gaussian noise with standard deviations ranging from 0.1 to 1.9 pixels. The dependency of the pairwise relationship estimation on the noise level is shown in Table 1. The threshold for the RANSAC method was set to 1.0 pixels. The results show that the method is robust against uniform Gaussian noise and the reprojection error raises slower than the noise level. Nevertheless, with higher noise levels less points are detected as inliers and the computation time becomes considerably longer.

7.2 Performance of the linear solution to the calibration

The second experiment was performed to test the performance of the linear calibration method of Section 6.1. The estimated pairwise positions and orientations from the previous experiment were used to perform the linear calibration. After calibrating the synthetic cameras we reconstructed the 3D points from the noisy image points and re-

projected them onto the camera planes. The mean euclidean distance between the image points and their reprojections, together with the improvements achieved by using bundle adjustment are shown in Table 2. It should be noted that the linear method produces high errors quite fast. Nevertheless, it gets near the desired minimum of the nonlinear cost function quite well and the bundle adjustment can recover the correct calibration up to the evaluated noise level.

7.3 Performance on real data

The third experiment uses real data extracted from a video sequence of 480 frames, recorded at 15 fps. Our sphere detection algorithm is applied to extract the midpoints of the sphere. These midpoints are then used to perform the full calibration. For reconstruction we use all cameras that observed a point. A visualization of the reconstructed virtual calibration object, i.e. the path of the marker object and the reconstructed camera setup are shown in Fig. 4. For evaluation we use the median reprojection error this time. This is because the data includes outlier which disturb the mean error computation. The results can be seen in Table 3. Except for camera 7 which has a very high error all cameras are calibrated reasonably well. As can be seen from experiment two, bundle adjustment is able to recover the calibration up to the noise level which is present in the data. This lets us conclude, that the remaining error cannot be further improved on by changing rotation and translation parameters. Instead this error must be attributed to errors in the internal calibration and errors due to the fact that we use the midpoints of the detected ellipses as correspondences which do not generally correspond to the projected midpoint of the sphere [HS97]. This introduces some bias which we plan to remove in the future.

camera number	median reprojection error in pixels
1	1.8665
2	2.8460
3	1.6047
4	2.3499
5	1.7885
6	2.7048
7	5.1755
8	2.4068

Table 3: Median reprojection error of the reconstruction from the linear solution to the calibration dependent on the noise level and after bundle adjustment (experiment 3).

8 Conclusions and Future Work

We have presented a flexible camera calibration system that can be used even if the cameras have no common field of view. It requires no especially manufactured calibration object. The method is robust against noise and lifts two restrictions imposed by earlier methods. These are:

- the scene does not have to be dark
- the cameras can be calibrated as long as their camera graph is connected

For the future we plan to include the computation of the real projected midpoints of the sphere instead of using the midpoints of the detected ellipses as correspondences. Furthermore, the detection of all cycles in the graph is computationally expensive. Since the search for all cycles is a breadth first search this search visits longer cycles later. Therefore, the search can be stopped after sufficiently many cycles have been found. The longer cycles are supposedly the less important ones. Additionally non-uniform Gaussian noise present in the midpoints can be taken into account using covariance matrices throughout the estimation. This becomes necessary if the cameras are not perfectly synchronized. Due to the movement of the ball, the projected midpoints are not the projections of the same 3D-point any more. This fact can be modeled as non-uniform noise with a higher variance of the data in the direction of the trajectory than in the direction normal to it.

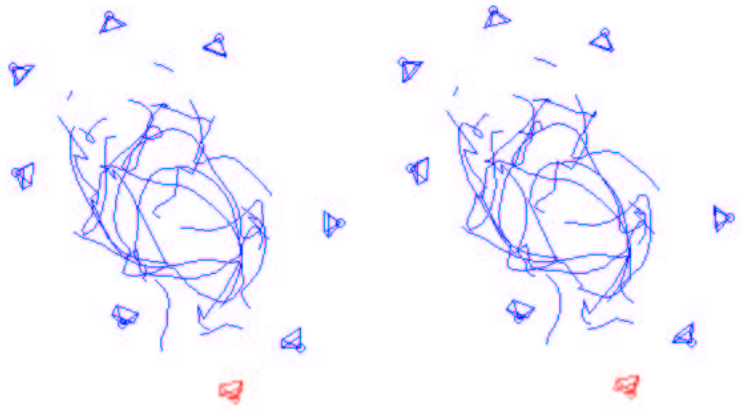

Figure 4: A top view of the reconstructed virtual calibration object and the resulting camera calibration. The lines show the path of the marker object, the pyramids depict the reconstructed camera positions. This is a cross-eye image that can be fused to give a depth impression.

References

[AP95] Ali Azarbayejani and Alex Pentland. Camera self-calibration from one point correspondence. Technical Report 341, MIT Media Lab, 1995.

[CDS00] Xing Chen, James Davis, and Philipp Slusallek. Wide Area Camera Calibration Using Virtual Calibration Objects. In *Proceedings of CVPR*, volume 2, pages 520–527, 2000.

[FB81] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6), June 1981.

[FL01] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. The MIT Press, 2001.

[Hor90] Berthold K.P. Horn. Recovering baseline and orientation from essential matrix. http://www.ai.mit.edu/people/bkph/papers/essential.pdf, January 1990.

[HS97] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *CVPR 97*, pages 1106–1112, 1997.

[HZ00] Richard Hartley and Andrew Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[KYD+03] Eun Jung Kim, Ki Hwan Yum, Chita R. Das, Mazin Yousif, and Jose Duato. Performance Enhancement Techniques for Infini-Band Architecture. In *HPCA-9*, pages 253–262, February 2003.

[NM65] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[Pol99] Marc Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, 1999.

[TLMS03] Christian Theobalt, Ming Li, Marcus Magnor, and Hans-Peter Seidel. A flexible and versatile studio for synchronized multi-view video recording. Research Report MPI-I-2003-4-002, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, April 2003.

[TMHF00] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[Tsa87] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), August 1987.

[Zha96] Zhengyou Zhang. A new multistage approach to motion and structure estimation: From essential parameters to euclidean motion via fundamental matrix. Technical Report RR-2910, INRIA, June 1996.