Tomáš Svoboda*
Daniel Martinec
Tomáš Pajdla
Faculty of Electrical Engineering
Czech Technical University
Prague
Czech Republic

*Correspondence to
svoboda@cmp.felk.cvut.cz

A Convenient Multicamera Self-Calibration for Virtual Environments

Abstract

Virtual immersive environments or telepresence setups often consist of multiple cameras that have to be calibrated. We present a convenient method for doing this. The minimum is three cameras, but there is no upper limit. The method is fully automatic and a freely moving bright spot is the only calibration object. A set of virtual 3D points is made by waving the bright spot through the working volume. Its projections are found with subpixel precision and verified by a robust RANSAC analysis. The cameras do not have to see all points; only reasonable overlap between camera subgroups is necessary. Projective structures are computed via rank-4 factorization and the Euclidean stratification is done by imposing geometric constraints. This linear estimate initializes a postprocessing computation of nonlinear distortion, which is also fully automatic. We suggest a trick on how to use a very ordinary laser pointer as the calibration object. We show that it is possible to calibrate an immersive virtual environment with 16 cameras in less than 60 minutes reaching about 1/5 pixel reprojection error. The method has been successfully tested on numerous multicamera environments using varying numbers of cameras of varying quality.

1   Introduction

With decreasing prices of powerful computers and cameras, smart multi-camera systems have started to emerge (Bobick et al., 1999; Brumitt, Meyers, Krumm, Kern, & Shafer, 2000; Trivedi, Mikic, & Bhonsle, 2000; Khan, Javed, Rasheed, & Shah, 2001; Svoboda, Hug, & Van Gool, 2002). A complete multicamera calibration is the inevitable step toward the efficient use of such systems even though many things can be accomplished with uncalibrated cameras in virtual environments and telepresence setups. To our best knowledge, no fully automatic calibration method for multicamera environments exists.

Very recent multicamera environments (Prince et al., 2002; Cheung, Baker, & Kanade, 2003), which are primarily designed for real-time 3D acquisition, use advanced calibration methods based on a moving plate (OpenCV, 2000; Zhang, 2000). These calibration methods do not require a 3D calibration object with known 3D coordinates. However, they share the main drawback with the old classical methods. The moving calibration plate is not visible in all cameras and the partially calibrated structures have to be chained together, a procedure very prone to errors. Kitahara et al. (2001) calibrated their large-scale multicamera environment by using a classical direct method (Tsai, 1987). The necessary 3D points are collected by a combined use of a calibration board and

a 3D laser-surveying instrument. Lee, Romano, and Stein (2000) established a common coordinate frame for a sparse set of cameras so that all cameras observe a common dominant plane. They tracked objects moving in this plane and from their trajectories they estimated the external parameters of the cameras in one coordinate system. Baker and Aloimonos (2003) proposed a calibration method for a multicamera network that requires a planar pattern with a precise grid.

We propose a fully automatic calibration method that yields complete camera projection models and requires only a small, easily detectable, bright spot. The bright spot can be created from a laser pointer by using a small trick. The user is required to wave the bright spot throughout the working volume. This is the only user action required. The projections of the bright spot are detected independently in each camera. We reach subpixel precision by fitting 2D Gaussian as a point-spread function. The points are validated through pairwise epipolar constraints. Projective motion and shape are computed via rank-4 factorization. Geometric constraints are applied and projective structures are stratified to Euclidean ones. The parameters of the nonlinear distortion are computed through iterative refinement. All these steps are described in this paper. The calibration software yields less than 1/5 pixel reprojection error even for cameras with significant radial distortion. The software is freely available.

Section 2 explains the mathematical theory behind the algorithm. Practical implementation of the algorithm is described in Section 3. Experiments on several different multicamera environments are presented in Section 4. The results are summarized in Section 5.

2 Algorithm—Theory

Let us consider m cameras and n object points $X_j = [X_j, Y_j, Z_j, 1]^T$, $j = 1, \cdots, n$. We assume the pinhole-camera model (see Hartley & Zisserman, 2000, for details). The 3D points $\mathbf{X}_j$ are projected to 2D image points $\mathbf{u}_j^i$ as
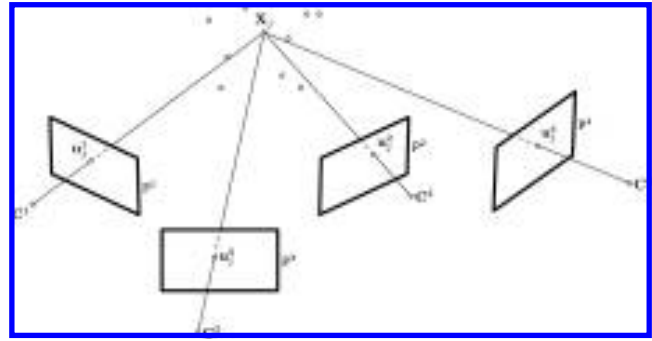

Figure 1. *Multicamera setup with four cameras.*

$$\lambda_j^i \begin{bmatrix} \mathbf{u}_j^i \\ v_j^i \\ 1 \end{bmatrix} = \lambda_j^i \mathbf{u}_j^i = P^i \mathbf{X}_j, \quad \lambda_j^i \in R^+ \tag{1}$$

where each P^i is a 3×4 matrix that contains 11 camera parameters, and u, v are pixel coordinates. A geometrical sketch of a four-camera setup is depicted in Figure 1. There are six parameters that describe camera position and orientation, sometimes called external parameters, and five internal parameters that describe the inner properties of the camera. $\mathbf{u}_j^i$ are observed pixel coordinates. The goal of the calibration is to estimate scales λ_j^i and the camera projection matrices P^i. We can put all the points and camera projections from Equation 1 into one matrix W_s:

$$W_s = \begin{bmatrix} \lambda_1^1 \begin{bmatrix} u_1^1 \\ v_1^1 \\ 1 \end{bmatrix} \cdots \lambda_n^1 \begin{bmatrix} u_n^1 \\ v_n^1 \\ 1 \end{bmatrix} \\ \vdots \quad \vdots \quad \vdots \\ \lambda_1^m \begin{bmatrix} u_1^m \\ v_1^m \\ 1 \end{bmatrix} \cdots \lambda_n^m \begin{bmatrix} u_n^m \\ v_n^m \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} P^1 \\ \vdots \\ P^m \end{bmatrix}_{3m \times 4} [\mathbf{X}_1 \cdots \mathbf{X}_n]_{4 \times n}$$

$$\tag{2}$$

$$W_s = PX, \tag{3}$$

where W_s is called the *scaled measurement matrix*, $P = [P^1 \cdots P^m]^T$ and $X = [\mathbf{X}_1 \cdots \mathbf{X}_n]$. P and X are referred to as the *projective motion* and the *projective shape*, respectively. If we collect enough noiseless points (u_j^i, v_j^i) and the scales λ_j^i are known, then W_s has rank 4

and can be factored into P and X (Sturm & Triggs, 1996). The factorization of Equation 3 recovers the motion and the shape up to a 4×4 projective transformation H:

$$W_s = PX = PHH^{-1}X = \hat{P}\hat{X}, \qquad (4)$$

where $\hat{P} = PH$ and $\hat{X} = H^{-1}X$. Any nonsingular 4×4 matrix may be inserted between P and X to get another compatible motion and shape pair $\hat{P}, \hat{X}$. The self-calibration process computes a matrix H such that $\hat{P}$ and $\hat{X}$ become Euclidean. This process is sometimes called *Euclidean stratification* (Hartley & Zisserman, 2000). The task of finding the appropriate H can be solved by imposing certain geometrical constraints. The most general constraint is the assumption that rows and columns of camera chips are orthogonal. Alternatively, we can assume that some internal parameters of the cameras are the same, which is more useful for a monocular camera sequence. The minimal number of cameras for a successful self-calibration depends on the number of known camera parameters, or on the number of parameters that are unknown but are the same for all cameras. For instance, eight cameras are needed when the orthogonality of rows and columns is the only constraint, and three cameras are sufficient if all principal points are known or if the internal camera parameters are completely unknown but are the same for all cameras (Hartley & Zisserman). We describe the Euclidean stratification in more detail in Section 2.2.

2.1 Projective Reconstruction by Factorization with Filling the Missing Points

Martinec and Pajdla's method (2002) was used for recovery of projective shape and motion from multiple images by factorization of a matrix containing the images of all scene points. This method can handle perspective views and occlusions jointly. The projective depths of image points are estimated by the Sturm and Triggs (1996) method using epipolar geometry. Occlusions are solved by the extension of the method by Ja-

cobs (1997) for filling the missing data. This extension can exploit the geometry of the perspective camera so that points with both known and unknown projective depths are used. The method is particularly suited for wide baseline multiple-view stereo.

It would be ideal to first compute the projective depths of all known points in W_s and then to fill all the missing elements of W_s by finding a complete matrix of rank 4 that would be equal (or as close as possible) to the rescaled W_s in all elements where W_s is known. Such a two-step algorithm is almost the ideal linearized reconstruction algorithm, which uses all data and has good statistical behavior. We have found that many image sets, in particular those resulting from wide baseline stereo, can be reconstructed in two steps. Otherwise, the two steps have to be repeated, while the measurement matrix W_s is not complete. In what follows, we shall describe the two steps of the algorithm.

2.1.1 Projective Depth Estimation.   We used Sturm and Triggs' method (1996) exploiting epipolar geometry but other methods may be applied too. Sturm and Triggs' method proposed two alternatives. The alternative with a central image is more appropriate for wide baseline stereo, while the alternative with a sequence is more appropriate for video sequences. In this paper, only the former alternative is explained (see Algorithm 1). For more details see Martinec and Pajdla (2002).

As noted in Sturm and Triggs (1996), any tree structure linking all images into a single connected graph can be used. This is especially advantageous when a large amount of occlusions is present in the data because then at least some depths can be recovered in each image and consequently all cameras can be estimated simultaneously. This modification will appear in a new version of the calibration package.

The algorithm for depth estimation using image c as the central image can be outlined as follows:

1. Set $\lambda_p^c = 1$ for all p's corresponding to known points $\mathbf{u}_p^c$.
2. For $i \neq c$ do the following: If images i and c have enough points in common to compute a funda-

mental matrix uniquely (see Martinec & Pajdla, 2002 for details) then compute the fundamental matrix F^{ic}, the epipole e^{ic}, and depths λ_p^t according to

$$\lambda_p^i = \frac{(e^{ic} \times u_p^i) \cdot (F^{ic} u_p^c)}{\|e^{ic} \times u_p^i\|^2} \lambda_p^c$$

if the right side of the equation is defined, where X stands for the cross-product.

2.1.2 Filling of Missing Elements in W_s.

The filling of missing data was first realized by Tomasi and Kanade (1992) for orthographic camera. Jacobs (1997) improved their method and we used our extension of his method for the perspective case. Often, not all depths can be computed because of missing data. Therefore, we extended Jacobs' method so that points with unknown depths are exploited also. First, the case where the depths of all points are known will be explained.

Jacobs (1997) treated the problem of missing elements in a matrix by fitting an unknown matrix of a certain rank to an incomplete noisy matrix resulting from measurements in images. Assume noiseless measurements, for a while, to make the explanation more simple. Assuming perspective images, an unknown complete $3m \times n$ matrix $\tilde{W}_s$ of rank 4 is fitted to W_s. Technically, a basis of the linear vector space that is spanned by the columns of $\tilde{W}_s$ is found.

Let the space generated by the columns of $\tilde{W}_s$ be denoted by β. Let β_t denote the linear hull of all possible fillings of the unknown elements of the tth four-tuple of columns of W_s which are linearly independent in coordinates known in all four columns. β is included in each β_t and thus also in their intersection, that is, $\beta \subseteq \cap_{t \in T} \beta_t$ where T is some set of indices. When the intersection is 4D, β is known exactly. If it is of a higher dimension, only an upper bound on β is known and more constraints from four-tuples must be added. Any column in $\tilde{W}_s$ is a linear combination of vectors of a basis of $\tilde{W}_s$. Thus, having a basis B of $\tilde{W}_s$, any incomplete column c in W_s containing at least four known elements, which in practice means six elements resulting from two known points, can be completed by finding the vector $\tilde{c}$

generated by B that equals c in the elements where c was known in W_s.

Because of noise in real data, the intersection $\cap_{t \in T} \beta_t$ quickly becomes empty. This is why β is searched for as the closest 4D space to spaces β_t in the sense of the minimal sum of square differences of known elements. More details are reported in Martinec and Pajdla (2002).

Recently, new constraints on the consistent set of all camera matrices were found. They are more robust to both significant camera movement and occlusions (Martinec & Pajdla, in press).

2.1.3 Filling of Missing Elements for Unknown Depths.

Jacobs' method (1997) cannot use image points with unknown depths. But matrix W_s constructed from measurements in perspective images often has many such points where the corresponding depths cannot be computed using the algorithm described in Section 2.1.1, due to occlusions. Therefore, we extended the method to exploit points with unknown depth in order to provide more and stronger constraints on the basis of the measurement matrix.

Let us first explain the extension for two images. Suppose that λ_p and u_p^i are known for $i = 1, 2$, and for $p = 1 \cdots 4$, except λ_4^2. Then, consider the first four columns of W_s to be the tth four-tuple of columns, A_t. A new matrix B_t, whose span will be denoted by β_t, can be defined using known elements of A_t as

$$A_t = \begin{bmatrix} \lambda_1^1 u_1^1 & \lambda_2^1 u_2^1 & \lambda_3^1 u_3^1 & \lambda_4^1 u_4^1 \\ \lambda_1^2 u_1^2 & \lambda_2^2 u_2^2 & \lambda_3^2 u_3^2 & ? u_4^2 \end{bmatrix} \rightarrow B_t$$

$$= \begin{bmatrix} \lambda_1^1 u_1^1 & \lambda_2^1 u_2^1 & \lambda_3^1 u_3^1 & \lambda_4^1 u_4^1 & 0 \\ \lambda_1^2 u_1^2 & \lambda_2^2 u_2^2 & \lambda_3^2 u_3^2 & 0 & u_4^2 \end{bmatrix}$$

It can be proven, that if B_t is of full rank (i.e., rank 5, here) then $\beta \subseteq Span(B_t)$, which is exactly the constraint on β. See Martinec and Pajdla (2002) for details on how to construct the matrix B_t in a general situation. By also including image points with unknown projective depths, the spaces β_t, spanned by four-tuples of columns become smaller; thus, solving the reconstruction problem becomes more efficient.

2.1.4 Combining the Filling Method with Depth Estimation.

Due to occlusions, the projective depth estimation can be carried out in various ways depending on which depths are computed first and if and how those already computed are used to compute the others. One way of depth estimation will be called a *strategy*. Depending on the strategy chosen, different subsets of depths are computed and different submatrices of W_s are filled. It may happen that when some strategy exploiting the epipolar geometry of some image pair is used, the fundamental matrix cannot be computed due to occlusions. Consequently, depths needed to form a constraint on β in one of the images cannot be estimated; thus the missing data in the image cannot be filled and the two steps of the depth estimation and filling have to be repeated.

From the structure of the missing data, it is possible to predict a good strategy for depth estimation that results in a good reconstruction. Some criterion on the quality of a strategy is needed. For scenes reconstructible in more steps, such a criterion also determines which subset of depths it is better to compute first.

The following two observations have been made: First, the more iterations performed, the less accurate are the results obtained, because the error from the former iteration spreads in subsequent iterations. Second, assuming the data is contaminated by random noise, unknown elements should not be computed from less data when they can be computed from more data, and thus more accurately due to the law of big numbers. For more details on choosing the best strategy for depth estimation see Martinec & Pajdla (2002).

2.2 Euclidean Stratification

Assume the projective factorization is complete. Here, we provide a simplified derivation on how to get a full camera calibration without measuring coordinates of any set of 3D points. Our stratification is based on the concept of the absolute conic (Hartley & Zisserman, 2000). Several possibilities for the derivation of the absolute conic constraint exist. In our implementation, we put the origin of the world frame to the centroid of the (unknown) reconstructed 3D Euclidean

points, which is the approach used in Han and Kanade (2000). However, the formulation where the origin of the world frame is in the first camera center (Pollefeys, Koch, & Van Gool, 1999; Hartley & Zisserman, 2000) is equivalent. We extend the notation used in the previous sections. As already mentioned, the Euclidean projection matrices contain internal parameters K^i and external camera parameters rotation R^i and translation $\mathbf{t}^i$,

$$\hat{P}^i = \mu^i K^i [R^i \ \mathbf{t}^i], \tag{5}$$

where μ^i is some nonzero scale, and

$$K^i = \begin{bmatrix} f^i & 0 & u_0^i \\ 0 & \alpha^i f^i & v_0^i \\ 0 & 0 & 1 \end{bmatrix}, R^i = \begin{bmatrix} \mathbf{i}^{i\mathrm{T}} \\ \mathbf{j}^{i\mathrm{T}} \\ \mathbf{k}^{i\mathrm{T}} \end{bmatrix}, \text{ and } \mathbf{t}^i = \begin{bmatrix} \mathbf{t}_x^i \\ \mathbf{t}_y^i \\ \mathbf{t}_z^i \end{bmatrix}.$$

Putting all the camera projections from Equation 5 together yields

$$\hat{P}_{3m \times 4} = [M_{3m \times 3} \ \mathbf{T}_{3m \times 1}], \tag{6}$$

where

$$M = [\mathbf{m}_x^1 \ \mathbf{m}_y^1 \ \mathbf{m}_z^1 \cdots \mathbf{m}_x^m \ \mathbf{m}_y^m \ \mathbf{m}_z^m]^{\mathrm{T}},$$

$$\mathbf{T} = [T_x^1 \ T_y^1 \ T_z^1 \cdots T_x^m \ T_y^m \ T_z^m]^{\mathrm{T}},$$

and

$$\mathbf{m}_x^i = \mu^i f^i \mathbf{i}^i + \mu^i u_0^i \mathbf{k}^i,$$

$$\mathbf{m}_y^i = \mu^i \alpha^i f^i \mathbf{j}^i + \mu^i v_0^i \mathbf{k}^i, \tag{7}$$

$$\mathbf{m}_z^i = \mu^i \mathbf{k}^i.$$

Similar formulas hold for elements of $\mathbf{T}$. The shape matrix is represented by

$$\hat{X} = \begin{bmatrix} \nu_1 \mathbf{s}_1 & \nu_2 \mathbf{s}_2 & \cdots & \nu_n \mathbf{s}_n \\ \nu_1 & \nu_2 & \cdots & \nu_n \end{bmatrix},$$

and

$$\mathbf{s}_j = [x_j \ y_j \ z_j]^{\mathrm{T}},$$

$$\hat{X}_j = [\nu_j \mathbf{s}_j^{\mathrm{T}} \ \nu_j]^{\mathrm{T}}.$$

We put the origin of the world frame into the centroid of the scaled 3D points.

$$\sum_{j=1}^{n} \nu_j \mathbf{s}_j = 0.$$
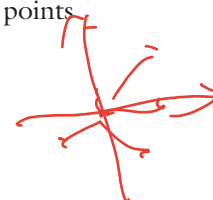
Expressing elements of the scaled measurement matrix W_s yields

$$\sum_{j=1}^{n} \lambda_j^i u_j^i = \sum_{j=1}^{n} (\mathbf{m}_x^i \nu_j \mathbf{s}_j + \nu_j T_x^i) = T_x^i \sum_{j=1}^{n} \nu_j. \quad (8)$$

Similarly

$$\sum_{j=1}^{n} \lambda_j^i \nu_j^i = T_y^i \sum_{j=1}^{n} \nu_j \quad \text{and} \quad \sum_{j=1}^{n} \lambda_j^i = T_z^i \sum_{j=1}^{n} \nu_j. \quad (9)$$

Let us define

$$H_{4\times4} = [A_{4\times3} \quad \mathbf{b}_{4\times1}], \quad (10)$$

putting Equations 10 and 6 into Equation 4 yields

$$[M \quad \mathbf{T}] = P[A \quad \mathbf{b}], \quad (11)$$

we have

$$T_x^i = \mathbf{P}_x^{iT} \mathbf{b}, \; T_y^i = \mathbf{P}_y^{iT} \mathbf{b}, \; T_z^i = \mathbf{P}_z^{iT} \mathbf{b}.$$

From Equations 8 and 9 we get

$$\frac{T_x^i}{T_z^i} = \frac{\sum_{j=1}^{n} \lambda_j^i u_j^i}{\sum_{j=1}^{n} \lambda_j^i} \quad \text{and} \quad \frac{T_y^i}{T_z^i} = \frac{\sum_{j=1}^{n} \lambda_j^i \nu_j^i}{\sum_{j=1}^{n} \lambda_j^i}.$$

Thus, we have $2m$ equations for the 4 unknown elements of $\mathbf{b}$.

From Equation 11,

$$MM^T = PAA^TP^T.$$

Define a new 4×4 symmetric matrix

$$Q = AA^T.$$

We show how to propagate the constraints on MM^T to the constraints on 10 unknown elements of Q in the case of unknown focal lengths.

We assume square pixels and principal points to be known. We can then transform the pixel points $\mathbf{u}_j^i$ and write

$$u_0^i = 0, \quad \nu_0^i = 0, \quad \text{and} \quad \alpha^i = 1.$$

We insert these assumptions into (7), which yields

$$\|\mathbf{m}_x^i\|^2 = \|\mathbf{m}_y^i\|^2, \quad (12)$$

$$\mathbf{m}_x^{iT}\mathbf{m}_y^i = 0,$$

$$\mathbf{m}_x^{iT}\mathbf{m}_z^i = 0,$$

$$\mathbf{m}_y^{iT}\mathbf{m}_z^i = 0,$$

We have $4m$ equations for 10 unknowns of Q, and therefore at least three cameras are needed for the self-calibration. Remember, we know that $MM^T = PQP^T$. Thus

$$\|\mathbf{m}_x^i\|^2 = \mathbf{P}_x^{iT}Q\mathbf{P}_x^i.$$

We proceed similarly for the other constrained elements of Equation 12. After some manipulation we can rewrite the constraints from equation 12 into a set of linear equations and solve them by using singular value decomposition (SVD). Once Q is estimated, we can recover the A matrix by rank-3 factorization.

Most modern cameras have square pixels. However, we can self-calibrate from three cameras with nonsquare pixels too. The first constraint from Equation 12 does not hold, since $\alpha_i \neq 1$, thus leaving only $3m$ constraints. However we can add one more constraint fixing one of the scales μ_i in Equation 7. Thus, for instance

$$\|\mathbf{m}_z^1\| = 1,$$

completing $3m + 1$ constraints.

Once are A and $\mathbf{b}$ estimated, we compose the stratification matrix $H = [A \; b]$. Then, the Euclidean shape $\hat{X} = H^{-1}X$ and the Euclidean motion $\hat{P} = PH$ are computed. Indeed, the knowledge of $\hat{P}$ is all we need to know for 3D reconstruction. However, sometimes it is useful to separate the external and internal parameters. We know that

$$\hat{P}^i = \mu^i[K^iR^i \quad K^i\mathbf{t}^i].$$

The first 3×3 submatrix of $\hat{P}^i$ may be decomposed into the orthonormal rotation matrix R^i and the upper triangular calibration matrix K^i by RQ matrix decompo-

sition. The position of the camera center may be then computed (see Hartley & Zisserman, 2000) as

$$\mathbf{C}^i = -\mathbf{R}^{iT}\mathbf{t}^i.$$

2.3 Estimation of the Nonlinear Distortion

Lenses with short focal lengths are often used in immersive environments to guarantee sufficient field of view. However, such lenses have significant nonlinear distortion, which has to be corrected for precise 3D computation. We propose a reliable procedure for estimating the distortion that needs no additional information and uses the linear estimate as the initial step.

The principle is as follows: First, reconstruct the calibration points by using the linear parameters and then feed these 3D–2D correspondences into a standard method for estimation of the nonlinear distortion. The linear self-calibration is then repeated with the corrected point coordinates. This estimate-and-refine cycle is repeated until the required precision is reached. The complexity of the distortion model, that is, the number of parameters to be estimated, gradually increases between the cycles. This iterative approach yields an average reprojection error of around 1/5 pixel, assuming a carefully synchronized set of multiple cameras.

In general, any calibration package can be used for estimation of the nonlinear distortion. We decided to apply a part of the Caltech camera calibration toolbox (Bouguet, 2004). Its Matlab codes are freely available and the estimated parameters are compatible with the OpenCV (2000) library, which is useful for eventual on-line distortion removal.

2.4 Critical Configurations of Points and Cameras

It is well known that there are *critical configurations* of cameras and points for which self-calibration is not possible, in principle. We do not go into theoretical details, we rather give some advice on how to avoid potential problems arising from this degeneracy.

First of all, the calibration points should fill up the working volume. This demand naturally disqualifies one of the degenerate configurations when all points are *coplanar* (Hartley, 2000). We should note that the coplanarity of all points not only makes the projective reconstruction ambiguous, it also makes the computation of epipolar geometry impossible (Hartley & Zisserman, 2000). Moreover, given $m \geq 3$ cameras, configuration is critical if all points and cameras lie in the intersection of two distinct ruled quadrics (Kahl, Hartley, & Åström, 2001). In practice, however, this may rarely happen.

Even when the projective structure and motion are estimated correctly there are still critical positions of cameras that make Euclidean stratification impossible. Such positions are called *critical motions* (Sturm, 1997; Kahl, Triggs, & Åström, 2000). If all cameras and lenses are the same we shall consider the critical motions for self-calibration with constant internal parameters (Sturm, 1997). In fact, in multicamera systems there are several critical motions we may get quite close to in practice: (a) rotation around parallel axes and arbitrary rotations, (b) orbital motion, (c) pure translations, and (d) planar motion (this also includes orbital motion). When the internal parameters of the cameras are different, the critical motions are a bit more obscure. The critical motions vary depending on the number of internal parameters we know in advance; however, we should try to avoid the following camera motions: (a) rotation with at most two distinct centers (twisted pair ambiguity); (b) motion on two conics whose supporting planes are orthogonal and where the optical axis is tangent to the conic at each position; (c) translation along the optical axis, with arbitrary rotations around the optical axis; or (d) motion with at most two viewing directions (orientation of optical axes). See Kahl et al. (2000) for a more thorough explanation.

It should be noted that there is one more important motion that is *not* critical for our self-calibration method but is critical for an alternative method based on Kruppa's equations. The method based on Kruppa's equations fails when optical centers of all cameras lie on a sphere and the optical axes pass through the sphere's center, a very natural situation in many multicamera systems (Sturm, 2000).

The section about critical configuration and motions

might be summarized with the following suggestions: To avoid numerical instability we should: (a) fill up the working volume with calibration points as completely as possible, avoiding coplanarity, and (b) vary the cameras as well as their positions and orientations as much as is reasonable. This first suggestion is clear and mostly satisfiable. The second suggestion typically narrows down to not having the cameras all coplanar or with parallel optical axes.

3 Algorithm—Practical Implementation

In the previous section, we have argued that the data matrix W containing the image points is the only input we need for the calibration. This matrix may contain some missing points; however, the more the matrix is full, the more accurate and stable the calibration results may be expected. Finding points $\mathbf{u}_j^i$ and establishing correspondences across many images, a process called *image matching*, is a difficult task. We overcome the problem by waving a slightly modified laser pointer through the working volume (see Figure 2). We attach a small piece of transparent plastic on the top of the laser pointer in order to get better visibility from different viewpoints. The very bright projections of the laser can be detected in each image with subpixel precision by fitting an appropriate point-spread function. These particular positions are then merged together over time, thus creating projections of a virtual 3D object. Our proposed self-calibration scheme can be outlined as follows:

1. Find the projections of the laser pointer in the images.
2. Discard misdetected points by pairwise RANSAC analysis (Fischler & Bolles, 1981).
3. Estimate projective depths λ_j^i and fill the missing points by the method described in Section 2.1.
4. Optimize the projective structure by using the Bundle Adjustment (Triggs, McLauchlan, Hartley, & Fitzgibbon, 1999), if applicable.
5. Perform the rank-4 factorization of the matrix W_s

Figure 2. *Our modification of a laser pointer. A small piece of transparent green or red plastic is attached to the laser pointer. The modification was invented in order to get better visibility from different viewpoints. However primitive a solution it is, it does the job very well.*

to get projective shape and motion (Hartley & Zisserman, 2000).

6. Upgrade the projective structures to Euclidean ones by the method described in Section 2.2.
7. Detect the remaining outliers by evaluating the 2D reprojection error. Remove them and repeat steps 3–6 until no outlier remains.
8. Estimate the parameters of the nonlinear distortion and repeat steps 2–7. Stop if the reprojection error is below the required threshold or if the number of iterations exceeds the maximum allowed.
9. Optionally, if some true 3D information is known, align the computed Euclidean structures with a world system.

It should be noted that the complicated scheme proposed above is rather conservative in rejecting misdetected points. Some validation steps may be left out when calibrating well-controlled setups.

3.1 Finding Corresponding Points

We need a rather robust method for finding points since it is not always possible to make the working volume completely dark. The camera room may have windows and glossy surfaces, thus making misdetection probable. The finding procedure has to be entirely automatic. Any user interaction is not an option because of the large number of images and cameras. However, it is assumed that the imaging conditions provide enough contrast between the bright spot and background. Our automatic finding procedure contains the following steps:

1. The mean image and the image of standard deviation is computed for each camera. These two images represent the static scene and the projections of the laser pointer are found by comparing the actual image with these two.

2. The differential image is computed by using the appropriate color channel depending on the color of the laser pointer. A threshold is set to 4/5 of the maximum of the differential image. The image is discarded if any of the following conditions hold:

   a. The number of pixels in the thresholded differential image is much higher than the expected LED size.

   b. The maximum of the differential image is less than five times the standard deviation in this pixel.

   c. The thresholded pixels are not connected, that is, they compose more than one blob.

   d. The eccentricity of the detected blob exceeds a predefined threshold. This condition is against motion blur.

3. The neighborhood of the detected blob is resampled to a higher resolution by using bicubic interpolation in order to reach subpixel accuracy and robustness against irregular blob shapes.

4. A 2D Gaussian is fitted to this interpolated subimage by 2D correlation to get the final position of the LED projection.

The detection sequence above is very robust and works well in very different multicamera setups. The color of the LED and the approximate expected size of the LED may vary for different setups. If the size is uncertain it is more robust to set it a bit bigger. In practice, this value turned out to be extremely stable. The desired subpixel accuracy may be also specified; however, 1/5 of a pixel is the reasonable maximum that should suffice for most cases. Some of the validation steps above may be skipped when the imaging environment is more controlled. The 2D correlation in Step 4 is the most computationally expensive operation. Steps 2–4 take about 100 ms together for one 640×480 image with the expected LED size of 7 pixels and a 1/3 subpixel accuracy on a 2 GHz PIV machine (highly vectorized Matlab code).

3.2 Discarding Misdetected Points

Even though the procedure described in the previous section is fairly robust, some false points may survive. When some glossy surfaces are present in the scene (e.g., glass walls), the reflection of the laser light might be detected instead of the direct projection. These outliers would spoil the projective reconstruction and have to be discarded in advance. There are two discarding steps: The first step is a robust pairwise computation of epipolar geometry and removal of points that lie too far from epipolar lines. This step clears the data at the very beginning of the whole process. The second step is an iterative loop that removes outliers by analyzing 2D reprojection error.

3.2.1 Finding Outliers in Image Pairs.   The image pairs are iteratively reselected according to the number of visible corresponding pairs. The points that were already detected as outliers are removed from the list of points found in these two cameras. The epipolar geometry is robustly computed via the RANSAC 7-point algorithm (Hartley & Zisserman, 2000). The initial tolerated distance from epipolar lines has to be preset by the user. The exact value of the threshold does not matter very much. It should not be too low when using lenses with significant radial distortion because it would discard too many good

but distorted points. Too high a value just adds a few more iterations in the subsequent discarding steps. Importantly, any value between 1 and 15 should do the job. We use 10 pixels, which works well for all our datasets, which include cameras with severe radial distortions. The initial threshold is iteratively decreased during the refinement steps (Section 2.3) as the camera models become more and more precise.

3.2.2 Finding Outliers in Reprojected Points.

The validation step based on epipolar geometry may fail to discard a misdetected projection if it lies along the epipolar lines. However, such a point can often be correctly reconstructed in 3D space from other (good) projections. If projected back to the cameras where it was misdetected it exhibits large 2D reprojection errors. Such problematic points are discarded from further computation.

There is no additional fixed threshold for deciding what is a large reprojection error and what is not large. The threshold is computed dynamically from the threshold preset for the RANSAC computation as well as from the mean and variance of the reprojection errors.

3.3 Euclidean Stratification

The stratification works rather well when reliable projective structures are estimated in the previous steps. We assume that cameras are different, have orthogonal rows and columns, no skew, square pixels, and we initialize the principal points to be in the image centers. It follows, from the counting argument (Hartley & Zisserman, 2000), that we need at least three cameras to perform the self-calibration. The resulting Euclidean projection matrices (five) may be decomposed into the internal and external parameters. The initial assumption about zero skew and known principal points is not used in the final decomposition. The stratification, without assuming known aspect ratios, is generally less robust and may occasionally fail in the case of somehow unbalanced input data. We had no camera with nonsquare pixels to perform tests with real data. However, this case was implemented, too.

3.4 Alignment with a World Coordinate System

The self-calibration yields the external camera parameters in an unknown world coordinate frame with the origin in the centroid of the point cloud. In practical applications, it is often desirable to have all parameters in some well-founded coordinate frame. For CAVE environments, for instance, we would like to have the $z = 0$ plane be coincident with the CAVE floor. Several different approaches might be applied. Scene objects with known dimensions and positions might be localized in image(s) and used for the alignment. However, an automatic localization of such objects might be difficult in practice. We offer an alternative way to do the alignment. We utilize the knowledge of the approximate camera positions. Since we know the physical dimensions of the CAVE construction, we can approximate the positions of the camera centers without actually measuring them. Precision in the range of several centimeters or even less precise is enough for a reliable alignment. We need to know at least three camera positions, while having more will increase the robustness. The cameras used must not lie on one line. The similarity transformation between the camera positions that are computed by the self-calibration and the desired ones is computed using the algorithm (Arun, Huang, & Blostein, 1987). The similarity transformation is then applied to all Euclidean structures.

The positions of the cameras may not be always available. We can often assume generally planar movement of the user and a complete alignment is not required. Sometimes, a "bird's eye view" of the overall arrangement is enough. A plane is fitted to the reconstructed point cloud made under the coplanarity assumption and then rotated to the desired orientation.

3.5 Issues in Estimation of the Nonlinear Distortion

The complexity of the nonlinear model gradually increases during the iteration. The iterative estimate-and-refine process is surprisingly stable. The
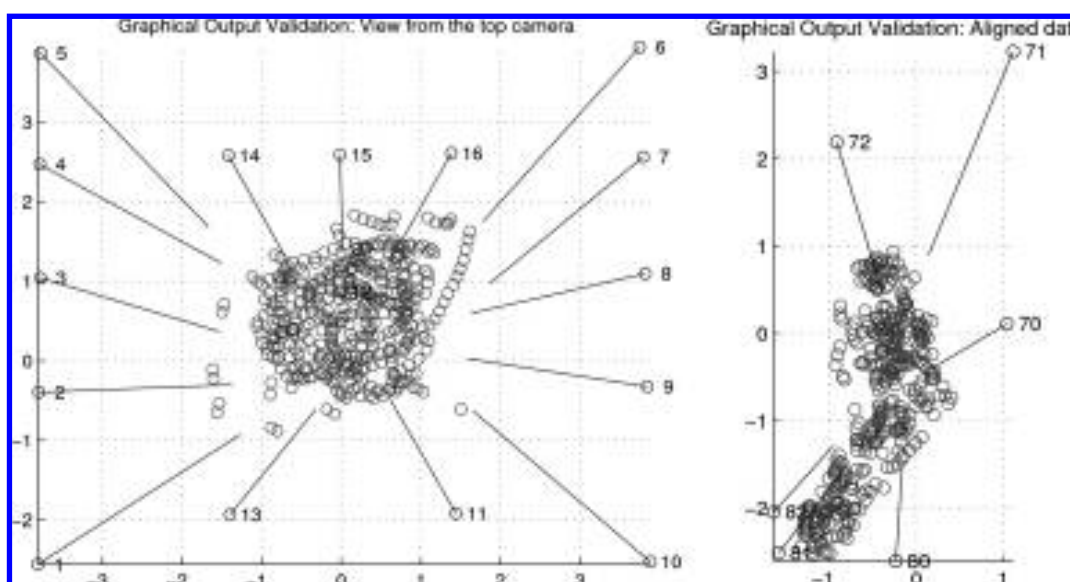
Figure 3. *Results of the Euclidean stratification for the Blue-C (left) and ViRoom (right) setups. Small circles with numbers denote positions of the camera centers, lines denote orientation of the optical axes. The clouds of circles show the reconstructed positions of the laser pointer. The parameters are aligned with the real world; dimensions are in meters.*

process may occasionally fail for cameras that have weak coverage of the image plane or too many outliers. To stabilize the estimation, it is sometimes better to decrease the complexity of the nonlinear model and disable the automatic increasing number of free parameters. A typical example is the estimation of the point of zero distortion. The estimation becomes unstable if the points are scattered on only one side of the image. It is better, in the case of such incomplete data, to disable the estimation of this parameter and put it into the image center. The final reprojection error may remain rather high, say about one pixel. However, wrongly estimating nonlinear parameters by overfitting could destroy the overall geometric consistency.

The filled 3D points are also used for the estimation. The number of iterations is by default constrained to 10. According to our experience, the whole refinement should converge within 5–6 iterations. If not, the desired model precision is perhaps set too optimistically, with respect to the quality of the data.

3.6 Validation of an Existing Calibration

Sometimes, we would like to know if the calibration is still valid or not. We may always recalibrate the system completely. However, this takes some time, and the resulting parameters will not be exactly the same as the old ones even if the setup remains the same. We suggest the following practical sampling approach:

1. Capture about 100 frames while waving the calibration object (bright spot).
2. Find the projections.
3. Perform a robust Euclidean reconstruction by trying all combinations of camera n-tuples, where n can be typically 2–4.
4. Select the camera n-tuple with the lowest reprojection error and its variance.
5. Evaluate the reprojection errors of this most consistent reconstruction.

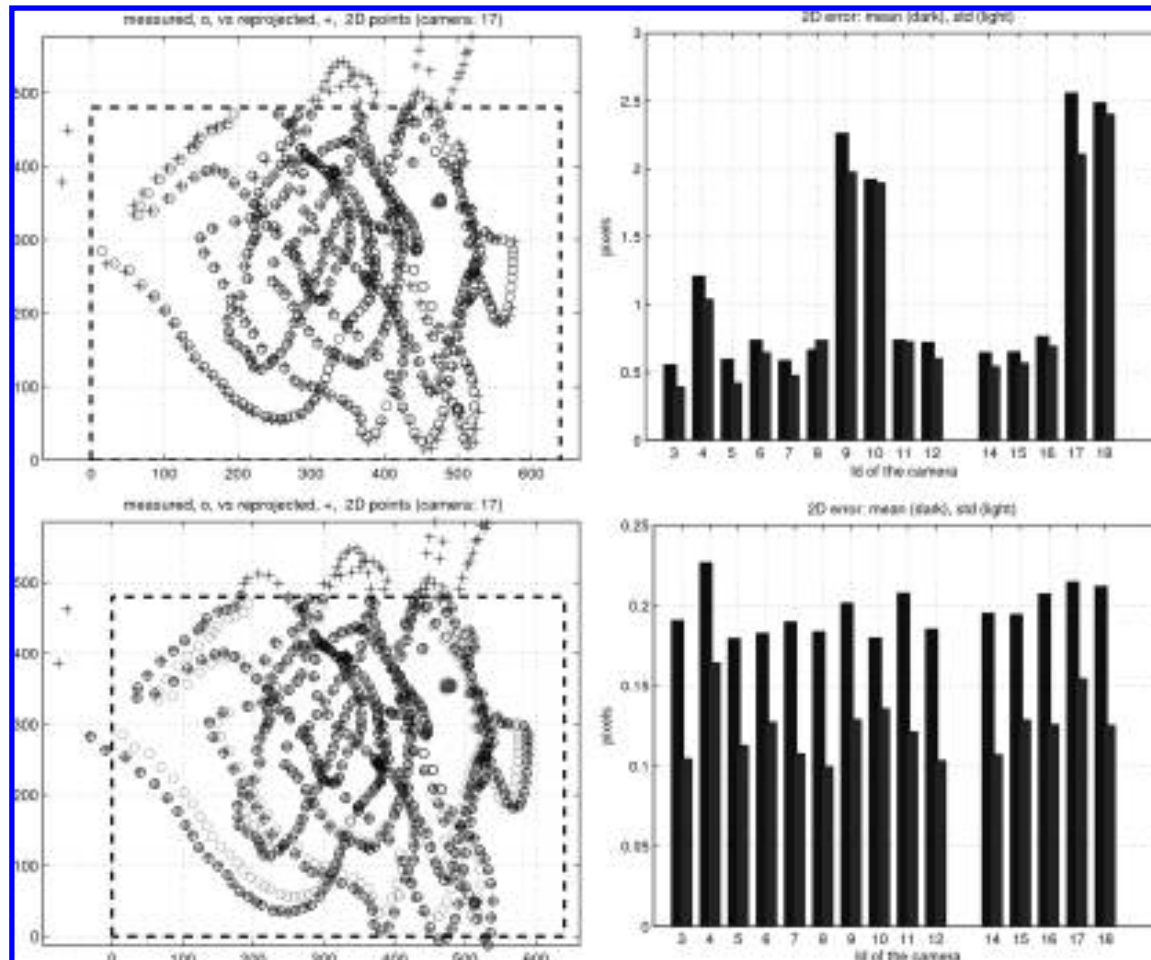The first two steps are the same as for the self-calibration itself. However, essentially, fewer points are

Figure 4. *Comparison of the linear model (top row) with the complete projection model including nonlinear distortion (bottom row). The left figures show the point reprojections in one of the cameras with significant radial distortion. The small circles denote the detected points. The crosses are back-projected reconstructed calibration points. The right figures show average reprojection errors and standard deviations in each camera. You can clearly distinguish Cameras 9, 10, 17, and 18, which are mounted inside CAVE and have the shortest lenses and thus significant distortion. Bottom row: Light circles are the originally detected points, dark ones show points after compensating for nonlinear distortion. Note also that the average reprojection errors decrease to around 1/5.*

required and there are no refinement loops and no bundle adjustment. Thus the complete validation may be completed in a few minutes.

4 Experiments

We would like to demonstrate two major features in which our solution outperforms competitors:

- The bright spot acting as a calibration device need not be visible in all cameras simultaneously.
- The parameters of the nonlinear distortion are estimated without any additional information.

The ability to fill missing points significantly broadens the possible application of our algorithm. Multiple cameras for immersive environments or telepresence virtual rooms often encompass the whole volume, thus posing
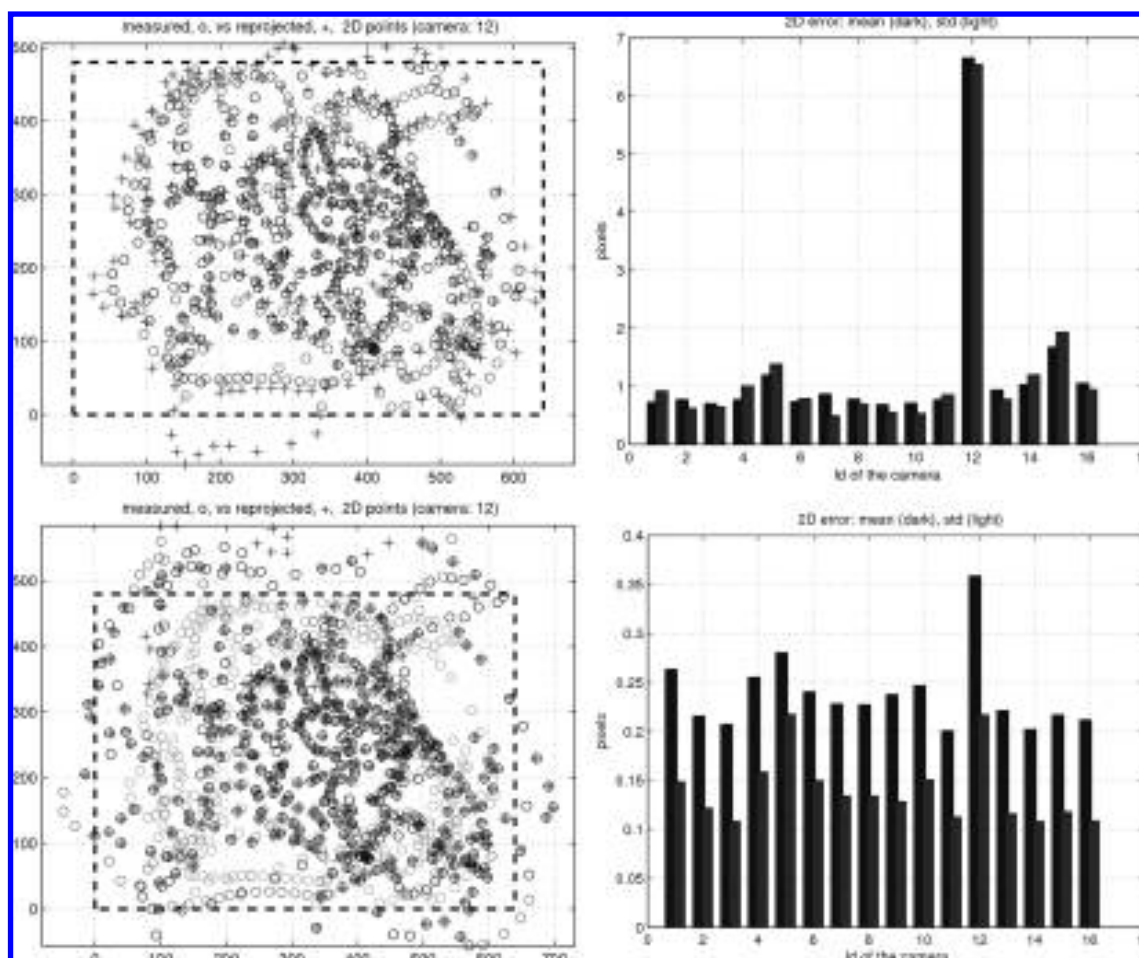
Figure 5. *Comparison of the linear model (top row) with the complete projection model including nonlinear distortion (bottom row). Example of an unbalanced multicamera system. Camera 12 has a fish-eye lens with huge radial distortion. After correction for nonlinear distortion, Camera 12 still has a higher reprojection error than the others. However, it decreased from the initial error of about 7 pixels to less than 0.4 pixels. The extreme radial distortion of the camera can clearly be recognized in the considerably different positions of the light (original points) and dark (undistorted points) circles.*

challenges in visibility. The Blue-C (Gross et al., 2003) setups, each with 16 cameras, have almost no occlusion because of a relatively empty working volume. Still, the calibration point is visible in all cameras in only a fraction of all calibration frames. Worse, points that are visible in all cameras usually span a small part of the possible working volume, thus making the estimation unstable. Occlusions and very different, or even disjoint, fields of view were common problems when using the mobile version of the ViRoom (Svoboda et al., 2002;

Doubek, Svoboda, & Van Gool, 2003) system. Calibration based only on the points visible in all cameras would be virtually impossible here. The filled points also take part in the estimation of the nonlinear distortion. Calibration results for both Blue-C and ViRoom setups are depicted in Figure 3.

We will show that our automatic estimation of the nonlinear lens parameters is able to compensate for a huge distortion in fish-eye lenses. This feature is necessary for very precise shape reconstruction applications.
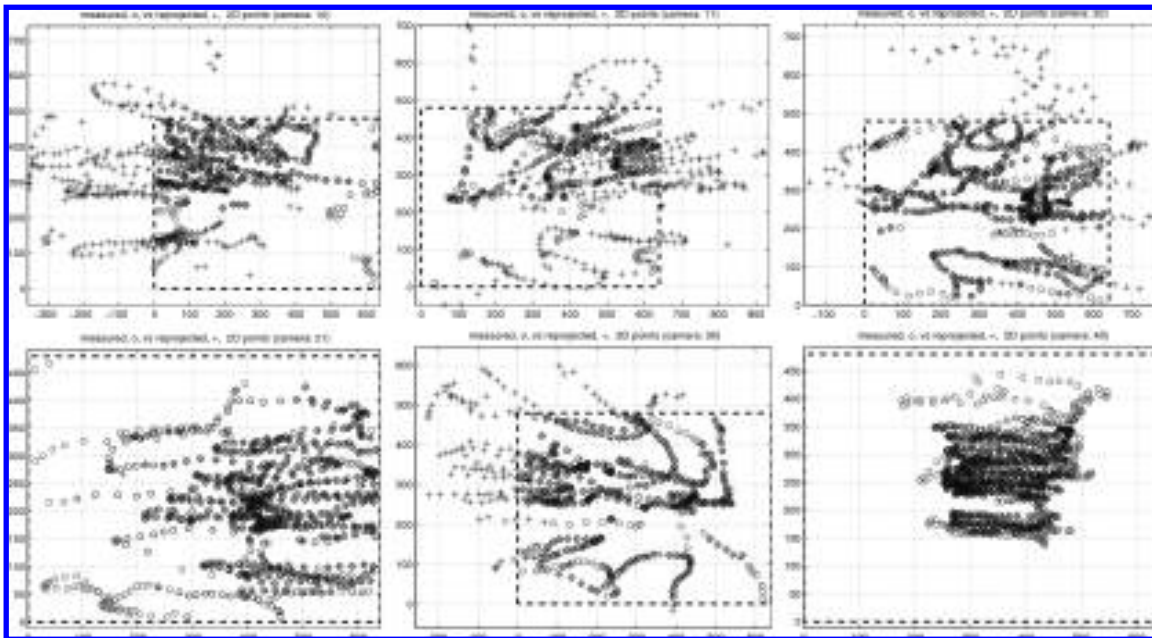
Figure 6. *Example of a less controlled multicamera setup. Note significant differences in the camera fields of view. The filled points essentially go outside the image planes (in graphs, denoted by the dashed rectangle). The last camera (bottom right) is quite far from the others and the points are clustered around the image center only. The first camera (top left) has a very unbalanced spread of points. Note also the considerable number of outliers caused by very difficult imaging conditions. The cameras are synchronized based on TCP/IP communication only. Nevertheless, the six-camera setup is reliably calibrated, with less than a 2-pixel reprojection error.*

We have used our algorithm on several multicamera setups scaling both quality and quantity of the cameras used. The two Blue-C setups have 16 cameras each. Firewire cameras are synchronized by an external sync signal; each camera has its own computer running under Linux for acquisition. The calibration sequences were acquired at 3–5 frames per second. The lower capturing frequency allows us to fill the working volume without accumulation of an unnecessarily high number of points. The speed of the waving is dictated by the shutter time of the cameras. It is desirable not to move very fast to avoid motion blur. The lenses span from 2.8 mm to 12 mm, exhibiting considerable radial distortion. Both Blue-C setups are used for high-quality reconstructions, which calls for a very high precision of the camera models. We show that we are able to calibrate the setups, achieving a reprojection error of about 1/5 pixel. The results for the Blue-C setups are summarized in Figures 4 and 5.

The ViRoom setups, both mobile and static, pose different challenges. Subpixel accuracy is not strictly required; 3D shape reconstruction is not the main application here. The setups are used for multicamera tracking, activity monitoring, and telepresence applications. The mobile version with six cameras and two laptops has been successfully used in a real factory environment. Both static and mobile setups can contain varying numbers of simple firewire cameras without external synchronization. One computer, a standard PC or a laptop running on Linux, often has to serve more than just one camera. The acquisition is synchronized via TCP/IP communication (Doubek et al., 2003), which is naturally far less precise than external synchronization by a HW system. The working volume often contains furniture and computers and cannot be completely darkened. The situation can be even worse. Frequently the camera fields of view only marginally overlap. Still, our system is able

to calibrate such setups with sufficient precision (see Figure 6). Estimation of the nonlinear distortion is difficult in such environments and may fail. It is typically necessary to fix the center of the nonlinear distortion to the image center.

5   Conclusion

A reliable scheme for a complete and fully automatic calibration of a multicamera network has been presented. A laser pointer or any similar bright-spot object is the only required additional hardware. Waving the object through the working volume is the only handwork requested. The object need not to be visible in all cameras. The nonlinear distortions are estimated from the same data set.

Experiments with different multicamera setups scaling quality and quantity demonstrated the broad usability of our algorithm.

Acknowledgments

References

Arun, K., Huang, T., & Blostein, S. (1987, September). Least-squares fitting of two 3-D point sets. *IEEE Transaction on Pattern Recognition and Machine Intelligence, 9*(5), 698–700.

Baker, P. T., & Aloimonos, Y. (2003). Calibration of a multi-camera network. In R. Pless, J. Santos-Victor, & Y. Yagi (Eds.). Proceedings of *Omnivis 2003: The 4th Workshop on Omnidirectional Vision and Camera Networks.*

Bobick, A., Intille, S., Davis, J., Baird, F., Pinhanez, C., Campbell, L., et al. (1999). The KidsRoom: A perceptually-based interactive and immersive story environment. *Presence: Teleoperators and Virtual Environments, 8*(4), 367–391.

Bouguet, J.-Y. (2004). *Camera calibration toolbox for Matlab.* Available from: http://www.vision.caltech.edu/bouguetj/calib_doc/.

Brumitt, B., Meyers, B., Krumm, J., Kern, A., & Shafer, S. (2000). EasyLiving: Technologies for intelligent environments. *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing,* 12–29.

Cheung, G. K., Baker, S., & Kanade, T. (2003). Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. *Proceedings of IEEE Computer Vision and Pattern Recognition.*

Doubek, P., Svoboda, T., & Van Gool, L. (2003). Monkeys—A software architecture for ViRoom — Low-cost multicamera system. In J. L. Crowley, J. H. Piater, M. Vincze, & L. Paletta (Eds.), *3rd International Conference on Computer Vision Systems* (pp. 386–395). Berlin: Springer.

Fischler, M., & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395.

Gross, M., Wuermlin, S., Naef, M., Lamboray, E., Spagno, C., Andreas, K., et al. (2003). Blue-c: A spatially immersive display and 3D video portal for telepresence. *ACM Transactions on Graphics (SIGGRAPH 2003), 22*(3), 819–827.

Han, M., & Kanade, T. (2000, December). Creating 3D models with uncalibrated cameras. *Proceedings of IEEE Computer Society Workshop on the Application of Computer Vision (WACV2000)*. Berlin: Springer.

Hartley, R. (2000). Ambiguous configurations for 3-view projective reconstruction. *European Conference on Computer Vision, 1,* 922–935.

Hartley, R., & Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge, UK: Cambridge University Press.

Jacobs, D. (1997). Linear fitting with missing data: Applica-

tions to structure from motion and to characterizing intensity images. *Computer Vision and Pattern Recognition,* 206–212.

Kahl, F., Hartley, R., & Åström, K. (2001). Critical configurations for *n*-view projective reconstruction. *IEEE Computer Vision and Pattern Recognition, 2,* 158–163.

Kahl, F., Triggs, B., & Åström, K. (2000). Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision, 13,* 131–146.

Khan, S., Javed, O., Rasheed, Z., & Shah, M. (2001). Human tracking in multiple cameras. *International Conference on Computer Vision,* 331–336.

Kitahara, I., Saito, H., Akimichi, S., Onno, T., Ohta, Y., & Kanade, T. (2001, June). Large-scale virtualized reality. *IEEE Computer Vision and Pattern Recognition, technical sketches.*

Lee, L., Romano, R., & Stein, G. (2000). Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 758–767.

Martinec, D., & Pajdla, T. (2002). Structure from many perspective images with occlusions. In A. Heyden, G. Sparr, M. Nielsen, & P. Johansen (Eds.), *Proceedings of the European Conference on Computer Vision* (Vol. 2, pp. 355–369). Berlin: Springer-Verlag.

Martinec, D., & Pajdla, T. (in press). 3D reconstruction by fitting low-rank matrices with missing data. *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR 2005).*

OpenCV. (2000). *Open source computer vision library.* Available from: http://www.intel.com/research/mrl/research/opencv/.

Pollefeys, M., Koch, R., & Van Gool, L. (1999). Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision, 32*(1), 7–25.

Prince, S., Cheok, A. D., Farbiz, F., Williamson, T., Johnson,

N., Billinghurst, M., et al. (2002). 3D live: Real time captured content for mixed reality. *International Symposium on Mixed and Augmented Reality (ISMAR'02)* (pp. 7–13). IEEE Press.

Sturm, P. (1997). Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. *IEEE Computer Vision and Pattern Recognition,* 1100–1105.

Sturm, P. (2000). A case against Kruppa's equations for camera self-calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(10), 1199–1204.

Sturm, P., & Triggs, B. (1996). A factorization based algorithm for multi-image projective structure and motion. *European Conference on Computer Vision* (p. 709–720). Berlin: Springer-Verlag.

Svoboda, T., Hug, H., & Van Gool, L. (2002). ViRoom—low cost synchronized multicamera system and its self-calibration. In L. Van Gool (Ed.), *Pattern Recognition, 24th DAGM Symposium* (pp. 515–522). Berlin: Springer.

Tomasi, C., & Kanade, T. (1992). Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision, 9*(2), 134–154.

Triggs, B., McLauchlan, P., Hartley, R., & Fitzgibbon, A. (1999). Bundle adjustment—A modern synthesis. In W. Triggs, A. Zisserman, & R. Szeliski (Eds.), *Vision algorithms: Theory and practice* (pp. 298–375). Berlin: Springer Verlag.

Trivedi, M. M., Mikic, I., & Bhonsle, S. K. (2000, June). Active camera networks and semantic event databases for intelligent environments. *IEEE Workshop on Human Modeling, Analysis and Synthesis (in conjunction with CVPR).*

Tsai, R. Y. (1987). A versatile camera calibration technique for high-accurancy 3D machine vision metrology using off-the-shelf cameras and lenses. *IEEE Journal of Robotics and Automation, RA-3*(4), 323–344.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(11), 1330–1334.