A Convenient Multi-Camera Self-Calibration for Virtual Environments

Article *in* Presence Teleoperators & Virtual Environments · August 2005

DOI: 10.1162/105474605774785325 · Source: DBLP

3 authors:

Tomas Svoboda
Czech Technical University in Prague
62 PUBLICATIONS **2,204** CITATIONS
SEE PROFILE

Daniel Martinec
unaffiliated
20 PUBLICATIONS **1,080** CITATIONS
SEE PROFILE

Tomas Pajdla
Czech Technical University in Prague
268 PUBLICATIONS **17,199** CITATIONS
SEE PROFILE

CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY

REPRINT

A Convenient Multi-Camera Self-Calibration for Virtual Environments

Tomáš Svoboda[1,2], Daniel Martinec[2] and Tomáš Pajdla[2]

svoboda@cmp.felk.cvut.cz

[1]Computer Vision Lab
Department of Information Technology and Electrical Engineering
Swiss Federal Institute of Technology
[2]Center for Machine Perception
Department of Cybernetics,
Czech Technical University in Prague

Tomáš Svoboda, Daniel Martinec, and Tomáš Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4), August 2005. To appear.

The final version of this article will be published in Presence, Vol. 14, Issue 4, published by The MIT Press.

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/svobodaPRESENCE2005.pdf

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

A Convenient Multi-Camera Self-Calibration for Virtual Environments

Tomáš Svoboda[1,2], Daniel Martinec[2] and Tomáš Pajdla[2]

[1]Computer Vision Lab

Department of Information Technology and Electrical Engineering

Swiss Federal Institute of Technology

[2]Centre for Machine Perception

Department of Cybernetics, Czech Technical University in Prague

svoboda@cmp.felk.cvut.cz

Revised version: June 16, 2004

Virtual immersive environments or telepresence setups often consist of multiple cameras which 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 sub-pixel 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 post-processing computation of non-linear 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 30 minutes reaching about 1/5 pixel reprojection error. The method has been successfully tested on numerous multi-camera environments with a varying number and quality of cameras used.

1 Introduction

With decreasing prices of powerful computers and cameras, smart multi-camera systems have started to emerge [4, 6, 17, 27, 30]. A complete multi-camera calibration is the inevitable step

towards 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 multi-camera environments, exists.

Very recent multi-camera environments [22] or [7], which are primarily designed for real-time 3D acquisition, use advanced calibration methods based on a moving plate [1, 32]. 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 whose procedure is very prone to errors. Kitahara et al., [18] calibrated their large scale multi-camera environment by using a classical direct method [31]. The necessary 3D points are collected by a combined use of a calibration board and a 3D laser-surveying instrument. Lee et al., [19] 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 [3] proposed a calibration method for a multi-camera network which requires a planar pattern with a precise grid.

We propose a fully automatic calibration method which 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 sub-pixel 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 non-linear 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 multi-camera environments are presented in Section 4. The results are shortly summarized in Section 5.

2 Algorithm — theory

Let us consider m cameras and n object points $\mathbf{X}_j = [X_j, Y_j, Z_j, 1]^\top, j = 1, \ldots, n$. We assume the pinhole camera model, see [12] for details. The 3D points $\mathbf{X}_j$ are projected to 2D image points $\mathbf{u}_j^i$ as

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

where each $\mathtt{P}^i$ is a 3×4 matrix that contains 11 camera parameters, and u, v are pixel coordinates. There are six parameters that describe camera position and orientation, sometimes called external parameters, and five internal parameters which 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

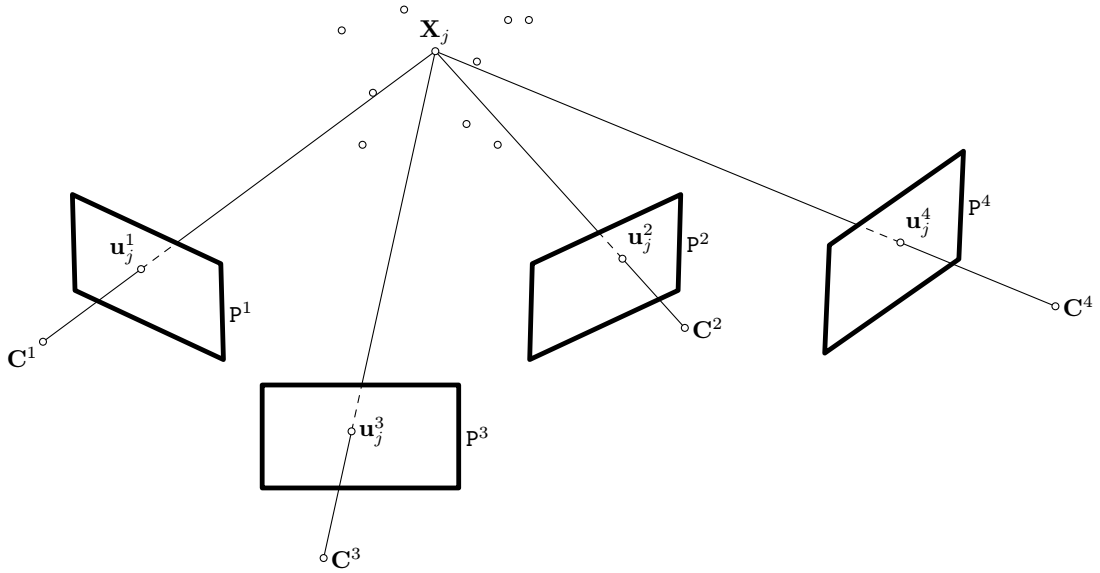

Figure 1: Multi-camera setup with 4 cameras.

projection matrices $\mathtt{P}^i$. We can put all the points and camera projections (1) into one matrix $\mathtt{W}_s$:

$$\mathtt{W}_s = \begin{bmatrix} \lambda_1^1 \begin{bmatrix} u_1^1 \\ v_1^1 \\ 1 \end{bmatrix} & \cdots & \lambda_n^1 \begin{bmatrix} u_n^1 \\ v_n^1 \\ 1 \end{bmatrix} \\ \vdots & \vdots & \vdots \\ \lambda_1^m \begin{bmatrix} u_1^m \\ v_1^m \\ 1 \end{bmatrix} & \cdots & \lambda_n^m \begin{bmatrix} u_n^m \\ v_n^m \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \mathtt{P}^1 \\ \vdots \\ \mathtt{P}^m \end{bmatrix}_{3m\times 4} [\mathbf{X}_1 \cdots \mathbf{X}_n]_{4\times n} \tag{2}$$

$$\mathtt{W}_s = \mathtt{P}\mathtt{X}\,, \tag{3}$$

where $\mathtt{W}_s$ is called the *scaled measurement matrix*, $\mathtt{P} = [\mathtt{P}^1 \cdots \mathtt{P}^m]^\top$ and $\mathtt{X} = [\mathbf{X}_1 \cdots \mathbf{X}_n]$. $\mathtt{P}$ and $\mathtt{X}$ 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 $\mathtt{W}_s$ has rank 4 and can be factored into $\mathtt{P}$ and $\mathtt{X}$ [26]. The factorization of (3) recovers the motion and the shape up to a 4×4 projective transformation $\mathtt{H}$:

$$\mathtt{W}_s = \mathtt{P}\mathtt{X} = \mathtt{P}\mathtt{H}\mathtt{H}^{-1}\mathtt{X} = \hat{\mathtt{P}}\hat{\mathtt{X}}\,, \tag{4}$$

where $\hat{\mathtt{P}} = \mathtt{P}\mathtt{H}$ and $\hat{\mathtt{X}} = \mathtt{H}^{-1}\mathtt{X}$. Any non-singular 4×4 matrix may be inserted between $\mathtt{P}$ and $\mathtt{X}$ to get another compatible motion and shape pair $\hat{\mathtt{P}}, \hat{\mathtt{X}}$. The self-calibration process computes such a matrix $\mathtt{H}$, that $\hat{\mathtt{P}}$ and $\hat{\mathtt{X}}$ become Euclidean. This process is sometimes called *Euclidean stratification* [12]. The task of finding the appropriate $\mathtt{H}$ 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, 8 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 [12]. We describe the Euclidean stratification in more detail in Section 2.2.

2.1 Projective reconstruction by factorization with filling the missing points

Martinec & Pajdla's method [20] 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 method of Sturm & Triggs [23] using the epipolar geometry. Occlusions are solved by the extension of the method by Jacobs [14] for filling the missing data. This extension can exploit the geometry of the perspective camera so that both points with known and unknown projective depths are used. The method is particularly suited for wide base-line multiple view stereo.

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

Projective depth estimation

We used Sturm & Triggs' method [23] exploiting the epipolar geometry but other methods may be applied too. The method [23] was proposed in two alternatives. The alternative with a central image is more appropriate for wide base-line 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 [20].

As noted in [23], 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.

Filling of missing elements in $\mathtt{W}_s$

The filling of missing data was first realized by Tomasi & Kanade [28] for orthographic camera. Jacobs [14] improved their method and we used our extension of his method for the perspective

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 fundamental matrix uniquely (see [20] for details) then compute the fundamental matrix $\mathtt{F}^{ic}$, the epipole $\mathbf{e}^{ic}$, and depths λ_p^i according to

$$\lambda_p^i = \frac{(\mathbf{e}^{ic} \times \mathbf{u}_p^i) \cdot (\mathtt{F}^{ic}\mathbf{u}_p^c)}{\|\mathbf{e}^{ic} \times \mathbf{u}_p^i\|^2}\lambda_p^c$$

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

Algorithm 1: Depth estimation using image c as the central image

case. Often, not all depths can be computed because of missing data. Therefore, we extended the method from [14], so that points with unknown depths are exploited also. At first, the case when the depths of all points are known will be explained.

Jacobs treated the problem of missing elements in a matrix as 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 $\tilde{\mathtt{W}}_s$ of rank 4 is fitted to $\mathtt{W}_s$. Technically, a basis of the linear vector space that is spanned by the columns of $\tilde{\mathtt{W}}_s$ is found.

Let the space generated by the columns of $\tilde{\mathtt{W}}_s$ be denoted by $\mathcal{B}$. Let $\mathcal{B}_t$ denote the linear hull of all possible fillings of the unknown elements of the t-th four-tuple of columns of $\mathtt{W}_s$ which are linearly independent in coordinates known in all four columns. $\mathcal{B}$ is included in each $\mathcal{B}_t$ and thus, also in their intersection, i.e. $\mathcal{B} \subseteq \bigcap_{t\in T} \mathcal{B}_t$ where T is some set of indices. When the intersection is 4D, $\mathcal{B}$ is known exactly. If it is of a higher dimension, only an upper bound on $\mathcal{B}$ is known and more constraints from four-tuples must be added. Any column in $\tilde{\mathtt{W}}_s$ is a linear combination of vectors of a basis of $\tilde{\mathtt{W}}_s$. Thus, having a basis $\mathtt{B}$ of $\tilde{\mathtt{W}}_s$, any incomplete column c in $\mathtt{W}_s$ 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 $\mathtt{B}$ which equals c in the elements where c was known in $\mathtt{W}_s$.

Because of noise in real data, the intersection $\bigcap_{t\in T} \mathcal{B}_t$ quickly becomes empty. This is why $\mathcal{B}$ is searched for as the closest 4D space to spaces $\mathcal{B}_t$ in the sense of the minimal sum of square differences of known elements. More details are reported in [20].

Recently, new constraints on the consistent set of all camera matrices were found. They are more robust to both significant camera movement and occlusions. The new method is to appear in an upcoming conference.

Filling of missing elements for unknown depths

Jacobs' method [14] cannot use image points with unknown depths. But, matrix $\mathtt{W}_s$ constructed from measurements in perspective images often has many such points where the corresponding depths cannot be computed using Algorithm 1, due to occlusions. Therefore, we extended the method to also exploit points with unknown depths 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^i and $\mathbf{u}_p^i$ are known for $i = 1, 2$, and for $p = 1 \dots 4$, except λ_4^2. Then, consider the first four columns of $\mathtt{W}_s$ to be the t-th four-tuple of columns, $\mathtt{A}_t$. A new matrix $\mathtt{B}_t$, whose span will be denoted by $\mathcal{B}_t$, can be defined using known elements of $\mathtt{A}_t$ as

$$\mathtt{A}_t = \begin{bmatrix} \lambda_1^1\mathbf{u}_1^1 & \lambda_2^1\mathbf{u}_2^1 & \lambda_3^1\mathbf{u}_3^1 & \lambda_4^1\mathbf{u}_4^1 \\ \lambda_1^2\mathbf{u}_1^2 & \lambda_2^2\mathbf{u}_2^2 & \lambda_3^2\mathbf{u}_3^2 & ?\ \mathbf{u}_4^2 \end{bmatrix} \longrightarrow \mathtt{B}_t = \begin{bmatrix} \lambda_1^1\mathbf{u}_1^1 & \lambda_2^1\mathbf{u}_2^1 & \lambda_3^1\mathbf{u}_3^1 & \lambda_4^1\mathbf{u}_4^1 & 0 \\ \lambda_1^2\mathbf{u}_1^2 & \lambda_2^2\mathbf{u}_2^2 & \lambda_3^2\mathbf{u}_3^2 & 0 & \mathbf{u}_4^2 \end{bmatrix}$$

It can be proven, that if $\mathtt{B}_t$ is of full rank (i.e. five, here) then $\mathcal{B} \subseteq Span(\mathtt{B}_t)$, which is exactly the constraint on $\mathcal{B}$. See [20] for details how to construct the matrix $\mathtt{B}_t$ in a general situation. By also including image points with unknown projective depths, the spaces $\mathcal{B}_t$, spanned by four-tuples of columns, become smaller, thus, solving the reconstruction problem becomes more efficient.

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 as well as 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 sub-matrices of $\mathtt{W}_s$ are filled. It may happen that when some strategy exploiting the epipolar geometry of some image pair is used, that the fundamental matrix cannot be computed due to occlusions. Consequently, depths needed to form a constraint on $\mathcal{B}$ 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 has 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 criterion also determines which subset of depths is better to be computed first.

The following two observations have been made: First, the more iterations performed, the results obtained are less accurate because the error from the former iteration spreads in subsequent iterations. Second, assuming the data is contaminated by a 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 [20].

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 [12]. 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 [11]. However, the formulation, where the origin of the world frame is in the first camera center [12, 21], is equivalent. We extend the notation used in the previous

sections. As already mentioned, the Euclidean projection matrices contain internal parameters $\mathtt{K}^i$ and external camera parameters, rotation $\mathtt{R}^i$ and translation $\mathbf{t}^i$,

$$\hat{\mathtt{P}}^i = \mu^i \mathtt{K}^i \begin{bmatrix} \mathtt{R}^i & \mathbf{t}^i \end{bmatrix} , \tag{5}$$

where μ^i is some non-zero scale, and

$$\mathtt{K}^i = \begin{bmatrix} f^i & 0 & u_0^i \\ 0 & \alpha^i f^i & v_0^i \\ 0 & 0 & 1 \end{bmatrix} , \quad \mathtt{R}^i = \begin{bmatrix} \mathbf{i}^{i\top} \\ \mathbf{j}^{i\top} \\ \mathbf{k}^{i\top} \end{bmatrix} , \quad \text{and } \mathbf{t}^i = \begin{bmatrix} t_x^i \\ t_y^i \\ t_z^i \end{bmatrix} .$$

Putting all the camera projections (5) together yields

$$\hat{\mathtt{P}}_{3m\times 4} = \begin{bmatrix} \mathtt{M}_{3m\times 3} & \mathbf{T}_{3m\times 1} \end{bmatrix} , \tag{6}$$

where

$$\begin{aligned} \mathtt{M} &= \begin{bmatrix} \mathbf{m}_x^1 & \mathbf{m}_y^1 & \mathbf{m}_z^1 & \cdots & \mathbf{m}_x^m & \mathbf{m}_y^m & \mathbf{m}_z^m \end{bmatrix}^\top , \\ \mathbf{T} &= \begin{bmatrix} T_x^1 & T_y^1 & T_z^1 & \cdots & T_x^m & T_y^m & T_z^m \end{bmatrix}^\top , \end{aligned}$$

and

$$\begin{aligned} \mathbf{m}_x^i &= \mu^i f^i \mathbf{i}^i + \mu^i u_0^i \mathbf{k}^i , \\ \mathbf{m}_y^i &= \mu^i \alpha^i f^i \mathbf{j}^i + \mu^i v_0^i \mathbf{k}^i , \\ \mathbf{m}_z^i &= \mu^i \mathbf{k}^i . \end{aligned} \tag{7}$$

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

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

and

$$\begin{aligned} \mathbf{s}_j &= \begin{bmatrix} x_j & y_j & z_j \end{bmatrix}^\top , \\ \hat{\mathbf{X}}_j &= \begin{bmatrix} \nu_j \mathbf{s}_j^\top & \nu_j \end{bmatrix}^\top . \end{aligned}$$

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

$$\sum_{j=1}^{n} \nu_j \mathbf{s}_j = \mathbf{0} .$$

Expressing elements of the scaled measurement matrix $\mathtt{W}_s$ yields

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

Similarly

$$\sum_{j=1}^{n} \lambda_j^i v_j^i = T_y^i \sum_{j=1}^{n} \nu_j \text{ and } \sum_{j=1}^{n} \lambda_j^i = T_z^i \sum_{j=1}^{n} \nu_j . \tag{9}$$

Let us define

$$\mathtt{H}_{4\times 4} = [\mathtt{A}_{4\times 3} \quad \mathbf{b}_{4\times 1}] \ , \tag{10}$$

putting (10) and (6) into (4) yields

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

we have

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

From (8, 9) we get

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

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

From (11),

$$\mathtt{M}\mathtt{M}^\top = \mathtt{P}\mathtt{A}\mathtt{A}^\top\mathtt{P}^\top \ .$$

Define a new 4×4 symmetric matrix

$$\mathtt{Q} = \mathtt{A}\mathtt{A}^\top \ .$$

We show how to propagate the constraints on $\mathtt{M}\mathtt{M}^\top$ to the constraints on 10 unknown elements of $\mathtt{Q}$ 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

$$u_0^i = 0 \ , \ v_0^i = 0 \ , \ \text{and} \ \alpha^i = 1 \ .$$

We insert these assumptions into (7) which yields

$$\begin{aligned} \|\mathbf{m}_x^i\|^2 &= \|\mathbf{m}_y^i\|^2 \ , \\ {\mathbf{m}_x^i}^\top \mathbf{m}_y^i &= 0 \ , \\ {\mathbf{m}_x^i}^\top \mathbf{m}_z^i &= 0 \ , \\ {\mathbf{m}_y^i}^\top \mathbf{m}_z^i &= 0 \ , \end{aligned} \tag{12}$$

We have $4m$ equations for 10 unknowns of $\mathtt{Q}$, and therefore at least three cameras are needed for the self-calibration. Reminder, we know that $\mathtt{M}\mathtt{M}^\top = \mathtt{P}\mathtt{Q}\mathtt{P}^\top$. Thus

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

Similarly for the other constrained elements of (12). After some manipulation we can rewrite the constraints (12) into a set of linear equations and solve them by using singular value decomposition (SVD). Once $\mathtt{Q}$ is estimated, we can recover the $\mathtt{A}$ matrix by rank-3 factorization.

Most modern cameras have square pixels. However, we can self-calibrate from three cameras with non-square pixels too. The first constraint from (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 (7). Thus, for instance

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

completing $3m + 1$ constraints.

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

$$\hat{\mathtt{P}}^i = \mu_i \left[\mathtt{K}^i\mathtt{R}^i \quad \mathtt{K}^i\mathbf{t}^i\right] .$$

The first 3×3 sub-matrix of $\hat{\mathtt{P}}^i$ may be decomposed into the orthonormal rotation matrix $\mathtt{R}^i$ and the upper triangular calibration matrix $\mathtt{K}^i$ by RQ matrix decomposition. The position of the camera centre may be then computed as

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

2.3 Estimation of the non-linear distortion

Lenses with short focal lengths are often used in immersive environments to guarantee sufficient field of view. However, such lenses have significant non-linear distortion which has to be corrected for precise 3D computation. We propose a reliable procedure for estimating the distortion which 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, i.e. 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 non-linear distortion. We decided to apply a part of the Caltech camera calibration toolbox [5]. Its Matlab codes are freely available and the estimated parameters are compatible with the OpenCV library [1] 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 the self-calibration is not possible, in principle. We do not go into theoretical details, we rather give some advice 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* [13]. We should note that the coplanarity of all points makes not only the projective reconstruction ambiguous it also makes the computation of epipolar geometry impossible [12]. Moreover, given $m \geq 3$ cameras, configuration is critical if all points and cameras lie in the intersection of two distinct ruled quadrics [15]. This may happen, however, hardly in practice.

Even though the projective structure and motion are estimated correctly there are still critical positions of cameras which make the Euclidean stratification impossible. Such positions are

called *critical motions* of cameras [16, 24]. If all cameras and lenses are the same we shall consider the critical motions for self-calibration with constant internal parameters [24]. In fact, in multi-camera systems there are several critical motions we may get quite close to in practice: (i) rotation around parallel axes and arbitrary rotations, (ii) orbital motion, (iii) pure translations, and (iv) planar motion (this also includes the 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: (i) rotation with at most two distinct centers (twisted pair ambiguity), or (ii) motion on two conics whose supporting planes are orthogonal and where the optical axis is tangent to the conic at each position, or (iii) translation along the optical axis, with arbitrary rotations around the optical axis, or (iv) motion with two viewing directions (orientation of optical axes) at most. See [16] for more thorough explanation.

It should be noted that there is one more important motion which 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, in the case where the optical centers of all cameras lie on a sphere and if the optical axes pass through the sphere's center, a very natural situation in many multi-camera systems [25].

The section about critical configuration and motions might be summarized in the following suggestions: To avoid numerical instability we should: (i) fill up the working volume with calibration points as completely as possible, avoiding coplanarity, (ii) and cameras as well as their positions and orientation shall be varied as much as is reasonable. This first suggestion is clear and mostly satisfiable. The second suggestion about cameras typically narrows down to not have the cameras all coplanar or with parallel optical axes.

3 Algorithm — practical implementation

In the previous section, we have argued that the data matrix $\mathtt{W}$ 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 sub-pixel 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 [9].
3. Estimate projective depths λ_j^i and fill the missing points by the method described in Section 2.1.

Figure 2: Immersive virtual environment BlueC [10] and our modification of a laser pointer. A small piece of transparent green or red plastic is attached to the laser pointer. The modification has been invented in order to get better visibility from different viewpoints. However primitive a solution it is, it does the job very well. The working volume is inside the glass CAVE. Four cameras are mounted on the top four corners of the construction and the remaining 12 cameras are mounted on the aluminum scaffold that encompasses the CAVE.

4. Optimize the projective structure by using the Bundle Adjustment [29], if applicable.

5. Perform the rank 4 factorization of the matrix W_s to get projective shape and motion [12].

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 non-linear distortion repeat the steps 2–7. Stop if the reprojection error is below the required threshold or if the number of iteration 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 I^i_μ and the image of standard deviation I^i_σ 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 5 times the standard deviation in this pixel.

   c) The thresholded pixels are not connected, i.e. 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 sub-pixel accuracy and robustness against irregular blob shapes.

4. A 2D Gaussian is fitted to this interpolated sub-image 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 multi-camera setups. The color of the LED and the approximate expected size of the LED may vary for different setups. If the size is not sure it is more robust to set the size a bit bigger. In practice, this value turned out to be extremely stable. The desired sub-pixel accuracy may be also specified however, 1/5 of a pixel is the reasonable maximum which 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 expected LED size 7 pixels and a 1/3 sub-pixel accuracy on a 2 GHz P IV 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: First step is a robust pairwise computation of epipolar geometry and removing 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 which removes outliers by analyzing 2D reprojection error.

3.2.1 Finding outliers in image pairs

The image pairs are iteratively re-selected 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 [12]. The initial tolerated distance from epipolar lines has to be pre-set 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 one and fifteen should do the job. We use ten 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 the 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 large and what is not large reprojection error. The threshold is computed dynamically from the threshold pre-set 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 centres. It follows, from the counting argument [12], that we need at least three cameras to perform the self-calibration. The resulting Euclidean projection matrices (5) 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 non-square 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 to 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 centres without actually measuring them. The precision in 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 used cameras must not lie on one line. The similarity transformation between the camera positions which are computed by the self-calibration and the desired ones is computed using the algorithm [2]. 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 not a complete alignment is 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 non-linear distortion

The complexity of the non-linear model gradually increases during the iteration. The iterative estimate and refine process is surprisingly stable. The process may occasionally fail for cameras which 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 non-linear 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 only on one side of the image. It is better to disable the estimation of this parameter and put it into the image center in case of such incomplete data. The final reprojection error may remain rather high, say about one pixel. However, wrongly estimated non-linear 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 though the setup remained the same. We suggest the following practical sampling approach:

1. Capture about 100 frames whilst 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, less points are required and there are no refinement loops and no bundle adjustment either. 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 needs not be visible in all cameras simultaneously.
- The parameters of the non-linear distortion are estimated without any additional information.

The ability of filling 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 challenges in visibility. Our Blue-C [10] setups each

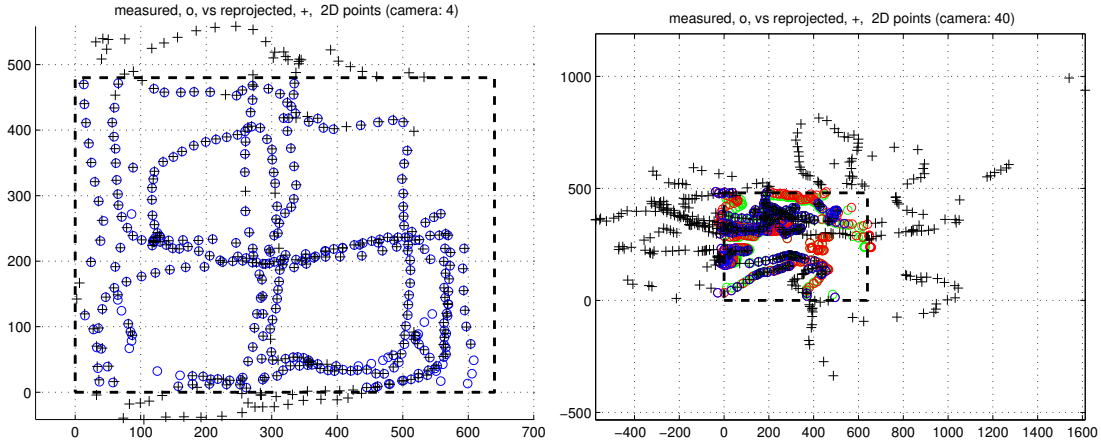

Figure 3: Filling "invisible" points. On the left, a Blue-C camera mounted on the ceiling. On the right, a camera from a ViRoom installation. The cameras have limited fields of view and do not see the whole working volume. The filling feature is clearly observable. Some points which have been reconstructed in 3D are clearly projected outside the image sensor (dashed line). They are visible in other cameras and filled into the measurement matrix.

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 which 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 our **Vi**Room [8, 27] 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 non-linear distortion.

We will show that our automatic estimation of the non-linear lens parameters is able to compensate for a huge distortion in the fish-eye lenses. This feature is necessary for very precise shape reconstruction applications.

We have used our algorithm on several multi-camera 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 unnecessary 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.

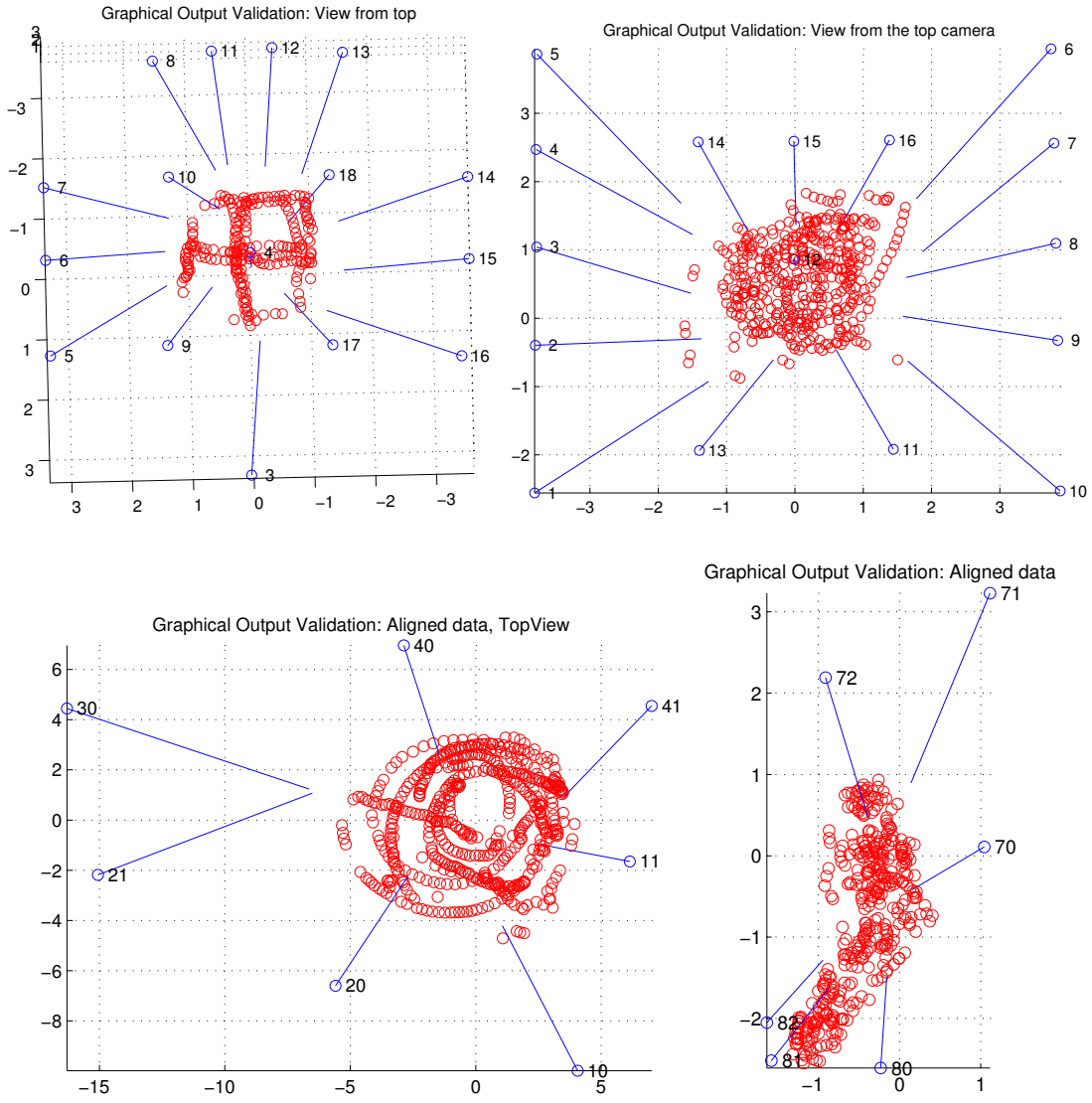

Figure 4: Results of the Euclidean stratification for the Blue-C (top row) and ViRoom (bottom row) setups. Small blue circles with numbers denote positions of the camera centers, blue lines denote orientation of the optical axes. The red circles show the reconstructed positions of the laser pointer.

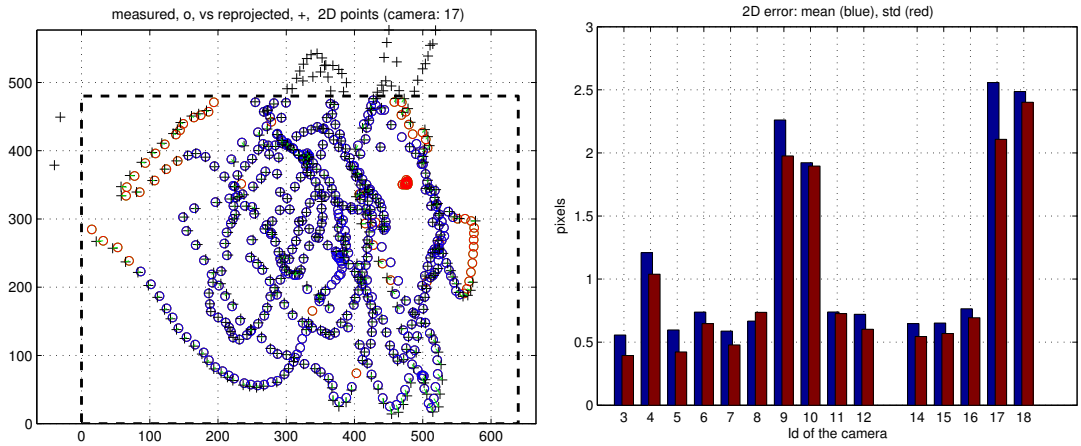

Figure 5: Results of the Linear model estimation. The left figure shows the point reprojections in one of the cameras with significant radial distortion. The small circles denote the detected points, the red ones are tentative outliers which were detected in the pairwise RANSAC validation. The crosses are back-projected reconstructed calibration points. The right figure shows average reprojection errors and standard deviations in each camera. You can clearly distinguish cameras No. 9, 10, 17, 18 which are mounted inside CAVE and have the shortest lenses and thus significant distortion.

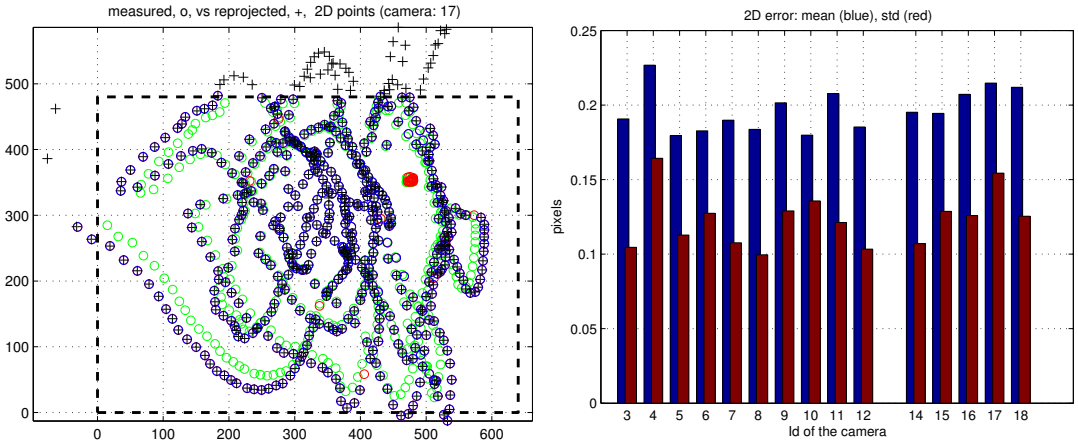

Figure 6: Complete projection model. The left figure shows the same camera as in the left figure in Fig 5. The green circles are the originally detected points, the blue ones show points after compensating for radial distortion. The right graph illustrates the well–balanced reprojection error of around $1/5$ a pixel. Compare with the reprojection of the linear model in Fig. 5.

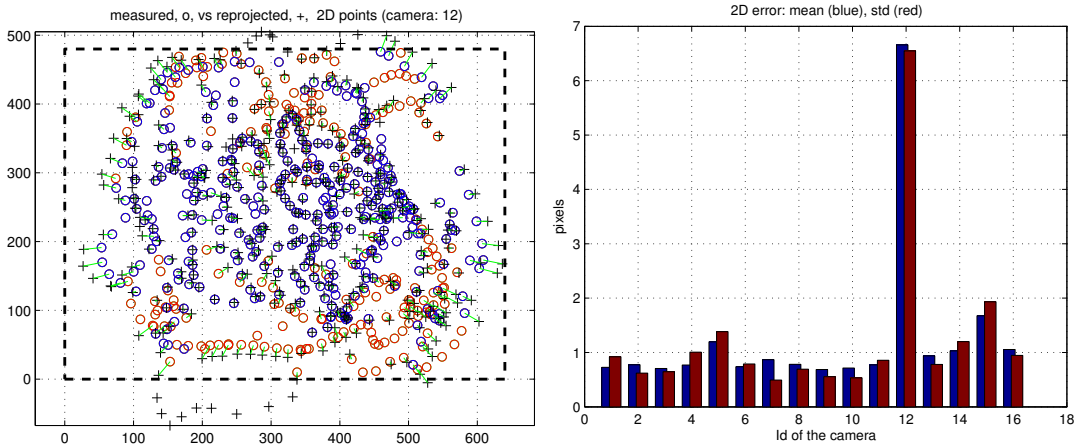

Figure 7: Results of the Linear model estimation. Example of an unbalanced multi-camera system. Camera 12 has a fish-eye lens with huge radial distortion.

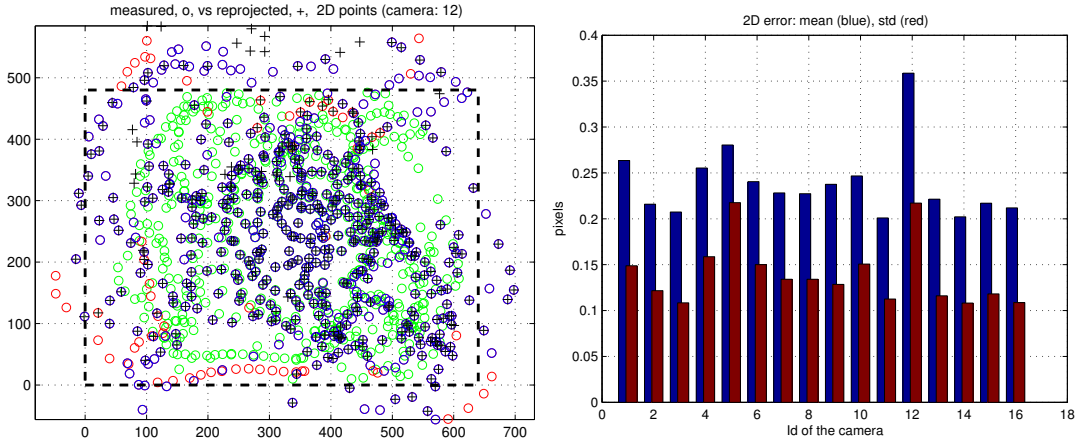

Figure 8: Results of the complete model estimation. Camera 12 still has a higher reprojection error than the others. However, from the initial error of about 7 pixels, it decreased to less than 0.4 pixels. The extreme radial distortion of the camera can be clearly recognized in considerably different positions of the green (original points) and blue/red circles (undistorted points).

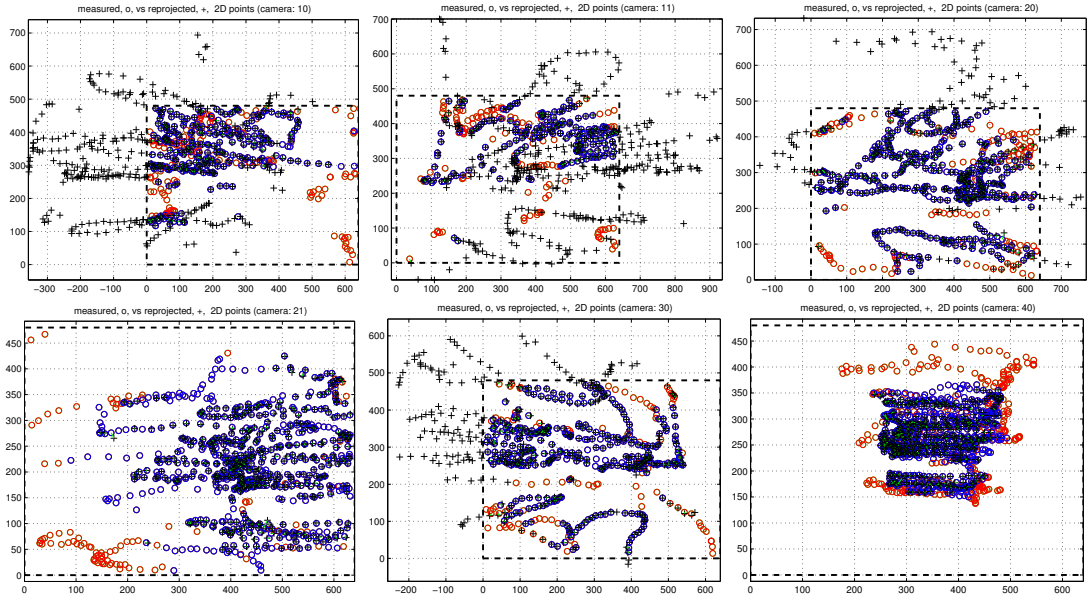

Figure 9: Example of a less controlled multi-camera setup. Note significant differences in the camera fields of view. The filled points essentially go outside the image planes (in graphs, denoted by the dashed rectangle). The last camera (bottom-right) is quite far from the others and the points are clustered around the image centre 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 6-camera setup is reliably calibrated, with less than a two pixel reprojection error.

The **Vi**Room setups, both mobile and static ones pose different challenges. Sub-pixel accuracy is not strictly required, 3D shape reconstruction is not the main application here. The setups are used for multi-camera 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 number 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 [8] which is naturally far less precise than external synchronization by a HW system. The working volume often contains furniture and computers and it 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. The estimation of the non-linear distortion is difficult in such environments and may fail. It is typically necessary to fix the centre of the non-linear distortion to the image centre. Simply speaking, you cannot get better precision of the calibration than your points are.

5 Conclusion

A reliable scheme for a complete and fully automatic calibration of a multi-camera 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 hand work required. The object needs not to be visible in all cameras. The non-linear distortions are estimated from the same data set.

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

Acknowledgements

We thank Ondřej Chum for his implementation of the 7-point RANSAC algorithm, Tomáš Werner for his Bundle Adjustment routines and Jean-Yves Bouguet for non-linear distortion codes. Student Dejan Radovic implemented a very first version of the stratification.

Tomáš Svoboda acknowledges the support of the Blue-C, poly-project of the Swiss Federal Institute of Technology and The Czech Academy of Sciences under project 1ET101210407. Daniel Martinec and Tomáš Pajdla ware supported by the grants GACR 102/01/0971, IST-2001-39184, MSMT Kontakt 22-2003-04, and The STINT under project Dur IG2003-2 062.

References

[1] Open source computer vision library. http://www.intel.com/research/mrl/research/opencv/. Last visited June 25, 2003.

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

[3] P. T. Baker and Y. Aloimonos. Calibration of a multicamera network. In R. Pless, J. Santos-Victor, and Y. Yagi, editors, *Omnivis 2003: Omnidirectional Vision and Camera Networks*, 2003.

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

[5] J.-Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/. Last visited June 17, 2003.

[6] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. Easyliving: Technologies for intelligent environments. In *Proceedings of the 2nd International Symposium on Hanheld and Ubiquitos Computing*, pages 12–29, September 2000.

[7] G. K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Computer Vision and Pattern Recognition*, 2003.

[8] P. Doubek, T. Svoboda, and L. Van Gool. Monkeys — a software architecture for ViRoom — low-cost multicamera system. In J. L. Crowley, J. H. Piater, M. Vincze, and L. Paletta, editors, *3rd International Conference on Computer Vision Systems*, number 2626 in LNCS, pages 386–395, Berlin, Germany, April 2003. Springer.

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

[10] M. Gross, S. Wuermlin, M. Naef, E. Lamboray, C. Spagno, K. Andreas, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, S. Kai, A. Vande Moere, and O. Staadt. Blue-c: A spatially immersive display and 3D video portal for telepresence. *ACM Transactions on Graphics (Siggraph 2003)*, 22(3):819–827, July 2003.

[11] M. Han and T. Kanade. Creating 3D models with uncalibrated cameras. In *Proceeding of IEEE Computer Society Workshop on the Application of Computer Vision (WACV2000)*, December 2000.

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[13] R. Hartley. Ambiguous configurations for 3-view projective reconstruction. In *European Conference on Computer Vision*, volume 1, pages 922–935, 2000.

[14] D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In *Computer Vision and Pattern Recognition*, pages 206–212, 1997.

[15] F. Kahl, R. Hartley, and K. Astrom. Critical configurations for n-view projective reconstruction. In *Computer Vision and Pattern Recognition*, volume 2, pages 158–163, 2001.

[16] F. Kahl, B. Triggs, and K. Aström. Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision*, 13:131–146, 2000.

[17] S. Khan, O. Javed, Z. Rasheed, and M. Shah. Human tracking in multiple cameras. In *International Conference on Computer Vision*, July 2001.

[18] I. Kitahara, H. Saito, S. Akimichi, T. Onno, Y. Ohta, and T. Kanade. Large–scale virtualized reality. In *Computer Vision and Pattern Recognition, Technical Sketches*, June 2001.

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

[20] D. Martinec and T. Pajdla. Structure from many perspective images with occlusions. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proceedings of the European Conference on Computer Vision*, volume II, pages 355–369, Berlin, Germany, May 2002. Springer-Verlag.

[21] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, August 1999.

[22] S. Prince, A. D. Cheok, F. Farbiz, T. Williamson, N. Johnson, M. Billinghurst, and H. Kato. 3D live: Real time captured content for mixed reality. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, pages 7–13. IEEE Press, September-October 2002.

[23] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conference on Computer Vision*, volume II, pages 709–720, 1996.

[24] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *Computer Vision and Pattern Recognition*, pages 1100–1105, June 1997.

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

[26] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conference on Computer Vision*, pages 709–720. Springer - Verlag, 1996.

[27] T. Svoboda, H. Hug, and L. Van Gool. V**i**Room — low cost synchronized multicamera system and its self-calibration. In L. Van Gool, editor, *Pattern Recognition, 24th DAGM Symposium*, number 2449 in LNCS, pages 515–522, Berlin, Germany, September 2002. Springer.

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

[29] B. Triggs, P. McLauchlan, R. Hartley, and A. 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, 1999.

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

[31] R. Y. Tsai. 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, August 1987.

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