Benno Heigl, Heinrich Niemann: Camera Calibration from Extended Image Sequences for Lightfield Reconstruction

Universität Erlangen–Nürnberg,
Lehrstuhl für Mustererkennung,
Martensstr. 3, 91058 Erlangen,
Germany
Email:
heigl@informatik.uni-erlangen.de

Workshop: Vision, Modeling and Visualization
Erlangen, November 1999

Pages: 43–50

Abstract:
This contribution treats the problem of camera calibration of extended image sequences with the goal to generate lightfields. Our method is mainly based on the factorization method by Sturm and Triggs which provides a closed–form solution if all projections of all scene points into all cameras are known. We extend the original method for extended image sequences, especially when scene points appear and disappear in the projections of the sequence, which is natural when moving a camera around an object. We apply the original method on parts of the sequence and merge the results together. As our experiments show, in spite of these extension, our method produces comparable results to the original method.

Camera Calibration from Extended Image Sequences for Lightfield Reconstruction

Benno Heigl *, Heinrich Niemann

Universität Erlangen–Nürnberg, Lehrstuhl für Mustererkennung,
Martensstr. 3, 91058 Erlangen, Germany
Email: heigl@informatik.uni-erlangen.de

Abstract

This contribution treats the problem of camera calibration of extended image sequences with the goal to generate lightfields. Our method is mainly based on the factorization method by Sturm and Triggs which provides a closed–form solution if all projections of all scene points into all cameras are known. We extend the original method for extended image sequences, especially when scene points appear and disappear in the projections of the sequence, which is natural when moving a camera around an object. We apply the original method on parts of the sequence and merge the results together. As our experiments show, in spite of these extension, our method produces comparable results to the original method.

1 Introduction

The goal of our work is to construct a Lightfield [5] from an uncalibrated image stream which is taken with a hand–held camera. For a realistic visualization many views all around a scene have to be taken. Therefore, we need a stable method for calibrating especially long image sequences.

Many investigations have been made to solve the problem of camera calibration when just knowing projections of scene points with unknown pose. Most approaches do a reconstruction from up to three successive images of the sequence and merge these reconstructions together by minimizing the total reprojection error of the reconstructed points (e.g. [1, 4]). They do not treat all data uniformly, but they sequentially reconstruct new camera poses with respect to those calibrated before. The whole reconstruction depends on the calibration of the first cameras.

For the case of orthographic projection, the first closed–form solution was given in [12]. There, a measurement matrix is built from projected image points and the camera positions as well as the scene points are reconstructed by a single factorization of this measurement matrix. Sturm and Triggs showed in [11] how to do a comparable solution for the case of perspective projection by building the measurement matrix from image projections and the corresponding projective depths. This method is very elegant as it recovers scene geometry and camera parameters for all images in one single step. The disadvantage of the closed–form solution is that it supposes that all scene points are visible throughout the whole image sequence. For real sequences this assumption does not hold.

In this contribution we focus on adapting the method [11] to sequences where points disappear and also new points appear. We apply the factorization method to parts of the sequence and merge them together. Doing this, we can benefit from the advantages of

*This work is partially funded by the German Research Foundation (DFG) under grant number SFB 603

the closed–form solution and also we are able to treat the problem of loosing points.

In all these methods for recovering structure from motion, in a first step the reconstruction is performed up to an unknown projective transformation. By applying self–calibration methods, constraints on the cameras can be introduced to get a reconstruction up to an unknown similarity transformation. We apply the linear method [8] to do self–calibration, which is also needed to merge together the partial solutions.

This article is structured as follows. In section 2, we introduce some notations. Section 3 gives a short description of the original factorization method of Sturm and Triggs and section 4 shows briefly how to do self–calibration with the method [8]. In section 5 we show how to apply these methods to real sequences. As in this step not all projected points can be considered, section 6 shows how the additional information given by these points can be used. Section 7 gives an evaluation of our method by testing it on simulated data.

2 Notation

In the following we presume some knowledge of projective geometry. An excellent introduction can be found in [7] which is focused on the special requirements for image analysis.

All vectors are denoted bold face and describe homogeneous vectors if not stated otherwise. A 3–D scene point is denoted by 4–vector $\boldsymbol{p}_k = (p_{k1}, p_{k2}, p_{k3}, p_{k4})^T$ corresponding to the Euclidean 3–vector $(p_{k1}/p_{k4}, p_{k2}/p_{k4}, p_{k3}/p_{k4})^T$, if the point is finite. Similar, a 2–D image coordinate of the projection of $\boldsymbol{p}_k$ into image number i is denoted by a homogeneous 3–vector $\boldsymbol{q}_{ik} = (q_{ik1}, q_{ik2}, q_{ik3})^T$. This implies that if a homogeneous vector is scaled by an arbitrary scalar not being zero, it does not change its Euclidean meaning.

The projection $\boldsymbol{q}_{ik}$ of $\boldsymbol{p}_k$ is achieved by a multiplication with a 3×4 projection matrix $\boldsymbol{P}_i$:

$$\lambda_{ik}\boldsymbol{q}_{ik} = \boldsymbol{P}_i\boldsymbol{p}_k \quad , \tag{1}$$

with $i = 1, \ldots, m$ and $k = 1, \ldots, n$. In a Euclidean framework, $\boldsymbol{P}_i$ can be factorized as $\boldsymbol{P}_i = \boldsymbol{K}_i\boldsymbol{R}_i^T(\boldsymbol{I}_3 \mid -\boldsymbol{t}_i)$, with $\boldsymbol{K}_i$ being an upper triangular calibration matrix and $\boldsymbol{R}_i$, $\boldsymbol{t}_i$ being a rotation matrix and a (Euclidean!) translation vector transforming the world coordinate system into the appropriate camera coordinate system.

If a variable is marked with a prime ($\boldsymbol{p}'$), it describes an estimation for the true value. If a variable is marked with a hat ($\widehat{\boldsymbol{p}}$), it describes a true value which is transformed projectively. If both markes are present ($\widehat{\boldsymbol{p}}'$), a variable describes a projectively transformed estimation of the true value.

3 Factorization Method

In this section we give a brief description of the factorization method of Sturm and Triggs [11]. Their main idea is to build a *measurement matrix* $\boldsymbol{W}$ and to solve the structure–from–motion problem by performing a single SVD (singular value decomposition, [9]) of $\boldsymbol{W}$.

To build the measurement matrix, the so–called *projective depths* λ_{ik} (see equation 1) have to be known. We suppose that all vectors $\boldsymbol{q}_{ik}$, $\boldsymbol{p}_k$ are scaled such that the last component is 1 and all $\boldsymbol{P}_k$ scaled such that their last row has norm 1. In this case the projective depths correspond to the orthogonal distances from the focal plane of each camera.

To recover these projective depths from projections only, fundamental matrices $\boldsymbol{F}_{ij}$ and epipoles $\boldsymbol{e}_{ij}$ between neighboring images i, j must be reconstructed. This can be achieved by applying known estimation techniques as described in [6]. In this contribution (following Sturm and Triggs) we apply the linear method of Hartley [3]. The projective depths can be updated from frame i to frame j by applying following equation:

$$\lambda_{ik} = \frac{(\boldsymbol{e}_{ij} \times \boldsymbol{q}_{ik})(\boldsymbol{F}_{ij}\boldsymbol{q}_{jk})}{\|\boldsymbol{e}_{ij} \times \boldsymbol{q}_{ik}\|^2}\lambda_{jk} \quad . \tag{2}$$

The projective depths of the first view can be chosen arbitrarily, e.g. $\lambda_{1k} = 1$.

Now all equations 1 for all scene points and all their projections are combined to one single matrix equation:

$$\boldsymbol{W} = \underbrace{\begin{pmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \vdots \\ \boldsymbol{P}_m \end{pmatrix}}_{\boldsymbol{\Pi}} \underbrace{(\boldsymbol{p}_1 \boldsymbol{p}_2 \dots \boldsymbol{p}_n)}_{\boldsymbol{\Psi}} \quad \text{with} \qquad (3)$$

$$\boldsymbol{W} = \begin{pmatrix} \lambda_{11}\boldsymbol{q}_{11} & \lambda_{12}\boldsymbol{q}_{12} & \dots & \lambda_{1n}\boldsymbol{q}_{1n} \\ \lambda_{21}\boldsymbol{q}_{21} & \lambda_{22}\boldsymbol{q}_{22} & \dots & \lambda_{2n}\boldsymbol{q}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m1}\boldsymbol{q}_{m1} & \lambda_{m2}\boldsymbol{q}_{m2} & \dots & \lambda_{mn}\boldsymbol{q}_{mn} \end{pmatrix} .$$

The estimation $\boldsymbol{W}'$ can be built knowing estimates for all projections of all scene points into all images and their corresponding projective depths. Note that the scale of each $\boldsymbol{p}_k$ and the scale of each $\boldsymbol{P}_i$ can be chosen arbitrarily if the λ_{ik} change accordingly. Therefore, to achieve a good numerical conditioning, each column c of $\boldsymbol{W}' = [w']_{rc}$ is rescaled such that $\sum_{r=1}^{3m} w'^2_{rc} = 1$ and each triplet of rows $(3i-2, 3i-1, 3i)$ is scaled such that $\sum_{c=1}^{n} \sum_{l=3i-2}^{3i} w'^2_{cl} = 1$.

An SVD is applied on $\boldsymbol{W}'$ resulting in the factorization $\boldsymbol{W}' = \boldsymbol{U}\,\mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_s)\,\boldsymbol{V}$. As the rank of $\boldsymbol{W}$ in the noise–free case is 4, we can, for the case having noise, recover the best rank 4 approximation (in the sense of least squares) by composing $\boldsymbol{W}'' = \boldsymbol{U}'\,\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)\,\boldsymbol{V}'$ with $\boldsymbol{U}'$ being the upper $3m \times 4$ sub–matrix of $\boldsymbol{U}$ and $\boldsymbol{V}'$ being the upper $4 \times n$ sub–matrix of $\boldsymbol{V}$. Knowing this factorization, estimates $\widehat{\boldsymbol{P}}'_i$ and $\widehat{\boldsymbol{p}}'_k$ can be found:

$$\widehat{\boldsymbol{\Pi}}' = \boldsymbol{U}'\boldsymbol{\Sigma} = \quad \text{and} \quad \widehat{\boldsymbol{\Psi}}' = \boldsymbol{\Sigma}\boldsymbol{V}' \qquad (4)$$

with $\boldsymbol{\Sigma} = \mathrm{diag}(\sqrt{\sigma_1}, \sqrt{\sigma_2}, \sqrt{\sigma_3}, \sqrt{\sigma_4})$. The hats shall show that this representation is not unique as for an arbitrary 4×4 transformation matrix $\boldsymbol{T}$ having rank 4 following equation holds: $(\widehat{\boldsymbol{P}}'_i \boldsymbol{T})(\boldsymbol{T}^{-1}\widehat{\boldsymbol{p}}'_k) = \widehat{\boldsymbol{P}}''_i \widehat{\boldsymbol{p}}''_k$, and similar $(\widehat{\boldsymbol{\Pi}}'\boldsymbol{T})(\boldsymbol{T}^{-1}\widehat{\boldsymbol{\Psi}}') = \widehat{\boldsymbol{\Pi}}''\widehat{\boldsymbol{\Psi}}''$. Therefore the structure of the scene and the projection matrices both can be reconstructed just up to an unknown projective transformation $\boldsymbol{T}$. This restriction is common to all reconstruction techniques if no additional knowledge on the properties of the cameras is available. To introduce such restrictions on the camera parameters, self–calibration methods can be applied as described in the following section.

4 Self–Calibration

As we have seen in the previous section, scene reconstruction can be done just up to an unknown projective transformation $\boldsymbol{T}$. The goal of self–calibration is to recover an estimation for matrix $\boldsymbol{T}'$ which maps the reconstructed scene points to estimations of the real scene points: $\boldsymbol{p}'_k = \boldsymbol{T}'^{-1}\widehat{\boldsymbol{p}}'_k \approx \boldsymbol{p}_k$ and accordingly $\boldsymbol{P}'_i = \widehat{\boldsymbol{P}}'_i\boldsymbol{T}' \approx \boldsymbol{P}_i$. This can be performed only by imposing some restrictions on the camera parameters.

In [14] the concept of the *absolute quadric* is introduced for doing self–calibration. The absolute quadric is described by the 4×4 matrix $\boldsymbol{\Omega} = \begin{pmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 \end{pmatrix}$. We restrict to an algebraic description and omit here the geometric interpretation of this concept as it is insubstantial for performing self–calibration. It is obvious that following equation holds:

$$\boldsymbol{P}_i\boldsymbol{\Omega}\boldsymbol{P}_i^T = \rho\boldsymbol{K}_i\boldsymbol{K}_i^T \quad , \qquad (5)$$

where $\boldsymbol{K}_i$ denotes the upper triangular calibration matrix. Suppose that the true scene points and camera matrices are transformed by $\boldsymbol{T}$. Then we see from Eq. 5 that $\boldsymbol{\Omega}$ is transformed to $\widehat{\boldsymbol{\Omega}} = \boldsymbol{T}\boldsymbol{\Omega}\boldsymbol{T}^T$. If $\boldsymbol{T}$ is a similarity transformation (meaning that $\boldsymbol{T} = \begin{pmatrix} \sigma\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{pmatrix}$ with $\boldsymbol{R}$ being a rotation matrix), the absolute quadric is invariant, therefore $\widehat{\boldsymbol{\Omega}} = \boldsymbol{\Omega}$. But in our case, $\boldsymbol{T}$ is an arbitrary projective transformation, and in general $\widehat{\boldsymbol{\Omega}} \neq \boldsymbol{\Omega}$. In [8] a linear method is presented to recover an estimate $\widehat{\boldsymbol{\Omega}}'$ uniquely for a given set of projection matrices, if all calibration matrices have the form $\boldsymbol{K}_i = \begin{pmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{pmatrix}$. When

the image skew, the principal point, and the aspect ratio is known, this can be achieved by transforming the measured image coordinates so that the principal point is $(0,0)$, the aspect ratio is 1, and the skew is 0. The only unknown camera parameter then is the focal length. The idea is to transform $\widehat{\boldsymbol{P}}'_1$ to be $(\boldsymbol{I}_{3\times 3}\,\mathbf{0})$:

$$(\boldsymbol{I}_{3\times 3}\,\mathbf{0}) = \widehat{\boldsymbol{P}}'_1 \underbrace{\begin{pmatrix} \widehat{\boldsymbol{P}}'_1 \\ \mathbf{0}^T\,1 \end{pmatrix}^{-1}}_{\boldsymbol{T}_n} \quad . \qquad (6)$$

All other projection matrices are transformed accordingly by multiplying $\widehat{\boldsymbol{\Pi}}'$ with $\boldsymbol{T}_n$. (For consistency, $\widehat{\boldsymbol{\Psi}}'$ is pre–multiplied with $\boldsymbol{T}_n^{-1}$.) In the noise–free case, following equation holds:

$$\boldsymbol{A}_i = \rho \begin{pmatrix} f_i^2 & 0 & 0 \\ 0 & f_i^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \widehat{\boldsymbol{P}}_i \widehat{\Omega} \widehat{\boldsymbol{P}}_i^T \quad . \qquad (7)$$

We see that these four equations have to be satisfied: $A_{i11} = A_{i22}, A_{i12} = A_{i13} = A_{i23} = 0$. Therefore having m frames, we get $4(m-1)$ equations for estimating the symmetric absolute quadric $\widehat{\boldsymbol{\Omega}}'$, which is parameterized by the 10 elements of its upper triangular sub–matrix. The focal length of each camera then can be calculated from $\boldsymbol{A}_i$. From $\widehat{\boldsymbol{\Omega}}'$, the plane at infinity can be retrieved as the nullspace of $\widehat{\boldsymbol{\Omega}}'$ (e.g. using SVD), and is denoted by the row–vector $\boldsymbol{p}'_\infty$. Supposing the camera coordinate system to coincide with the world coordinate system, we know that the unknown matrix $\boldsymbol{T}'$ must transform $(\boldsymbol{I}_{3\times 3}\,\mathbf{0})$ to $(\boldsymbol{K}'_1\,\mathbf{0})$. We also know that $\boldsymbol{p}'_\infty$ must be transformed to the standard plane at infinity $(0\,0\,0\,1)$. Applying these restrictions, the matrix $\boldsymbol{T}'$ is determined uniquely:

$$\boldsymbol{T}' = \begin{pmatrix} \boldsymbol{K}'_1 & \mathbf{0} \\ -\bar{\boldsymbol{p}}'_\infty \cdot \boldsymbol{K}'_1 & 1 \end{pmatrix} \quad , \qquad (8)$$

with $\bar{\boldsymbol{p}}'_\infty$ being the 3–vector consisting of the first three components of $\boldsymbol{p}'_\infty$ divided by its fourth component. The scale of the last row of $\boldsymbol{T}$ can be chosen freely. This is equal to scaling the whole scene with a unique factor which remains unknown.

5 Application to Real Image Sequences

The main disadvantage of the factorization method (section 3) is that all scene points must be visible throughout the whole sequence. In the analysis of real image sequences, this restriction is not practicable, because scene parts appear and disappear in dependence on the actual viewpoint and also there is no tracking algorithm which guarantees not to loose any scene points although they are visible. Hence we have to modify the algorithm to apply it to real environments.

Factorization of windows. During tracking point features in extended image sequences (for example, using the method [10]), it is usual to select a given number of point features (e.g. 1000) and to track them as long as possible. If in a particular frame features are lost, new features are selected to complete the number of tracked features. Therefore the factorization method cannot be applied to the whole sequence. We can inspect "windows" of frames where enough features could be tracked completely. On the one hand, the size of each window should not be too small, as the reconstruction error reduces with increasing number of frames (compare to the results in [11]). On the other hand, the window should not be too large, as the number of points which could be tracked throughout the whole window decreases and therefore the reconstruction is more sensitive to errors in the trajectory corresponding to a single scene point. The windows can be chosen to be a complete partitioning of the frames or they can be chosen to overlap. In the latter case, also the camera centers of the overlapping frames can be matched to each other. This is reasonable especially then, when the scene is compact and the cameras move around the scene with a large distance. Which choice performs best, depends on the particular situation. Let $\widehat{\boldsymbol{\Psi}}_v$ denote the matrix of the transformed scene points for window number v, according to the notation used in equation 3. $\mathcal{I}_v$ denotes the set of corresponding indices

of scene points which are used for this reconstruction in window.

Merging windows. If we have calculated a projective reconstruction for each window, we must link these reconstructions to each other. As each reconstruction is done up to an unknown projective transformation, there also exists an unknown transformation $\boldsymbol{T}_{vw}$ between two windows v and w which maps the transformed scene points of window v to the corresponding scene points of window w. Therefore these scene points must be visible in both windows. Let $\mathcal{I}_{vw} = \mathcal{I}_v \cap \mathcal{I}_w$ denote the set of indices of scene points which are visible in both windows v and w. Formally, for each scene point with an index $k \in \mathcal{I}_{vw}$, the matrix $\boldsymbol{T}_{vw}$ should satisfy the equation

$$\rho \, {}^{w}\widehat{\boldsymbol{p}}_k = \boldsymbol{T}_{vw} \, {}^{v}\widehat{\boldsymbol{p}}_k \quad , \tag{9}$$

where ${}^{v}\widehat{\boldsymbol{p}}_k$ denotes the scene point k in the v–th window. It is a column vector of $\widehat{\boldsymbol{\Psi}}_v$. If the vectors ${}^{w}\widehat{\boldsymbol{p}}_k$ and ${}^{v}\widehat{\boldsymbol{p}}_k$ are normalized such that their last component is 1, the scale ρ and the fourth component of the vector equation 9 can be eliminated. Therefore each correspondence of estimated scene points leads to three rows of a matrix:

$$\begin{pmatrix} {}^{w}\widehat{\boldsymbol{p}}'^{T}_{k} & \boldsymbol{0}^T & \boldsymbol{0}^T & -{}^{v}\widehat{p}'_{k1} \, {}^{w}\widehat{\boldsymbol{p}}'^{T}_{k} \\ \boldsymbol{0}^T & {}^{w}\widehat{\boldsymbol{p}}'^{T}_{k} & \boldsymbol{0}^T & -{}^{v}\widehat{p}'_{k2} \, {}^{w}\widehat{\boldsymbol{p}}'^{T}_{k} \\ \boldsymbol{0}^T & \boldsymbol{0}^T & {}^{w}\widehat{\boldsymbol{p}}'^{T}_{k} & -{}^{v}\widehat{p}'_{k3} \, {}^{w}\widehat{\boldsymbol{p}}'^{T}_{k} \end{pmatrix} .$$

The approximative nullspace of this matrix can be solved using SVD and describes the vector being the concatenation of the column vectors of $\boldsymbol{T}'_{vw}$.

In the noise–free case, this merging procedure works. But having errors in the reconstruction, the result is numerically not stable, because there are minimized distances within a projectively transformed scene. As tiny distances may become huge after a projective transformation and vice versa, the particular distances don't have a meaningful sense. Hence we have to minimize these distances within an Euclidean framework; we have to apply the self–calibration before merging.

In general, it is enough to do the self–calibration for reconstruction of the first window and to calculate the transformation from each window to the previous one. With this procedure the Euclidean framework is updated from one window to the next. But if there are small errors in the self–calibration of the first frame, the whole reconstruction will depend on this error. A final self–calibration step may be applied to improve the results.

Post–processing. The approach described above retrieves each camera matrix $\boldsymbol{P}'$ up to an unknown scale factor ρ. As $\boldsymbol{P}$ can be represented by $\boldsymbol{K}$, $\boldsymbol{R}$, and $\boldsymbol{t}$ having fixed scales, we have to determine ρ to convert $\boldsymbol{P}'$ to this representation: $\boldsymbol{P}' \rightarrow \boldsymbol{P}'/\rho$. For each projection matrix $\boldsymbol{P}'$, we can determine the oriented optical axis $\boldsymbol{a}' = (P'_{31} \, P'_{32} \, P'_{33})$ which is a Euclidean directional vector and is collinear with the third row of matrix $\boldsymbol{R}$. Therefore we know that $\|\rho\| = \|\boldsymbol{a}'\|$. The optical center $\boldsymbol{t}'$ can be determined uniquely from $\boldsymbol{P}'$. As the reconstructed points must lie in front of the camera, the scalar product of the directional vector $\boldsymbol{a}'$ with the vector $\boldsymbol{p}'_k - \boldsymbol{t}'$ should be positive for any visible scene point number k. To achieve this, we can determine $\rho = \mathrm{sgn}(\boldsymbol{a}'^{T}(\boldsymbol{p}'_k - \boldsymbol{t}'))\|\boldsymbol{a}'\|$.

6 Considering All Points

Up to now we have reconstructed all projection matrices and the 3–D position of some scene points. But there remain points which are visible in fewer frames than the window lengths and therefore have not been considered yet. Knowing all projection matrices, we are able to reconstruct these scene points. We can again formulate this request as a linear optimization problem. Knowing the projection $\boldsymbol{q}_{ik} = (q_{ik1}, q_{ik2}, q_{ik3})^T$ of the scene point number k in the image i, following equation holds:

$$\left(\begin{pmatrix} q_{ik1} \\ q_{ik2} \end{pmatrix} \boldsymbol{\psi}_{i3} - \begin{pmatrix} \boldsymbol{\psi}_{i1} \\ \boldsymbol{\psi}_{i2} \end{pmatrix} \right) \boldsymbol{p}_k = 0 \quad , \tag{10}$$

where $\boldsymbol{\psi}_{ij}$ denotes the j–th row–vector of matrix $\boldsymbol{P}_i$. The scale of $\boldsymbol{q}_{ik}$ has to be chosen such that its third component is 1. Writing this equation for each estimated projection of an unknown scene point and concatenating the

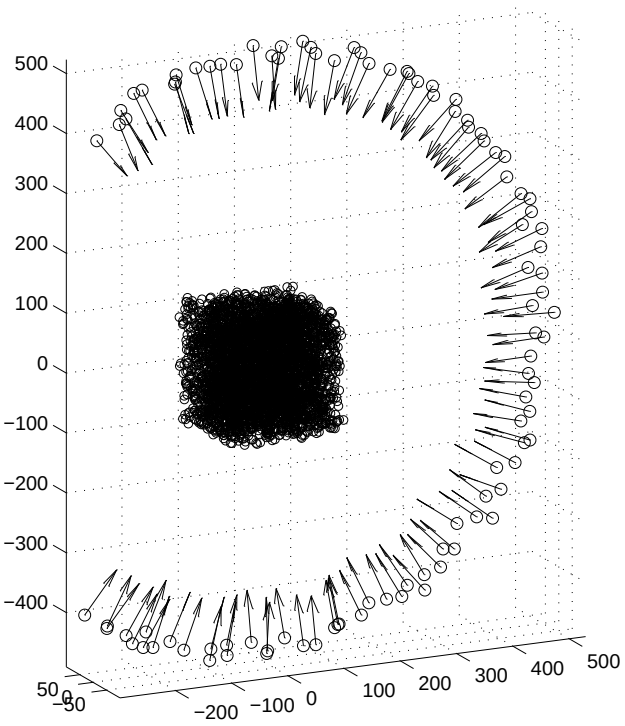

Figure 1: Example for a simulated test configuration. The arrows show the camera centers with their viewing directions; the points in the middle show the scene points. Note the small displacements of neighboring cameras.

left matrix of upper equation, we get a matrix whose nullspace describes the coordinates of the searched scene point $\boldsymbol{p}'_k$. The approximative nullspace again can be determined by applying SVD.

Another way to consider the remaining points is to complete the measurement matrix during each factorization step. As we have estimated the fundamental matrices between succeeding frames, we can predict their projections from three succeeding views using the method [2]. For this approach, many degenerate and nearly degenerate cases exist. E.g. the projection centers of three views must not be collinear, or the corresponding scene point must not lie on the plane built by the three projection centers. Therefore, the better way is to estimate the trifocal tensor (e.g. by method [13]) and to use that for prediction.

As small errors in the measured image coordinates as well as in the estimated fundamental matrices or trifocal tensors affect the predictions, this method of completing the measurement matrix should just be applied to fill small gaps. Otherwise, the reconstruction will depend too much on these erroneous estimated predictions.

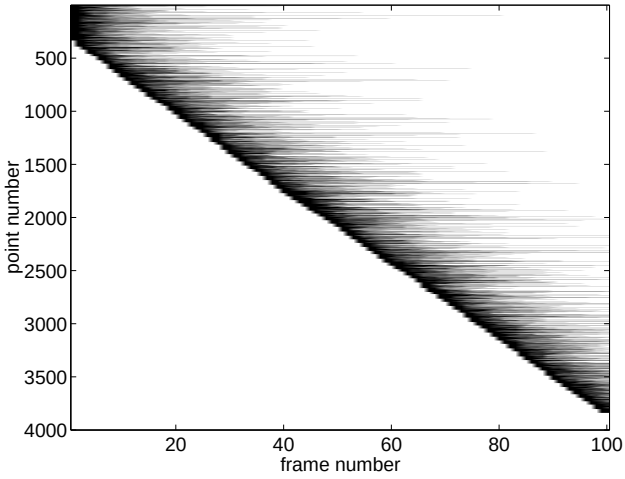

Figure 2: Simulated visibilities

7 Experiments

Simulation of camera movement and scene points. The scene points have been chosen to be randomly distributed in a cube with an edge length of 200. The camera moves around the center of the cube with a radius of 500. Its position as well as its orientation is perturbed randomly to avoid degeneracies of self–calibration (see section 4). This perturbation is also natural for sequences taken with a hand–held camera. The focal lengths have been chosen randomly in the range $800 \ldots 1200$. Figure 1 shows an example configuration. The scene points have been projected into each camera. To each projection, an error has been added with a uniform distribution within the range $-r \ldots r$. We call r the *noise level*.

Simulation of visibility. To simulate the visibility of each point, we made the assumption that the probability p for loosing a point in a frame is constant. Then we get a geometric distribution for the length t of the visibility for each point. Following equation

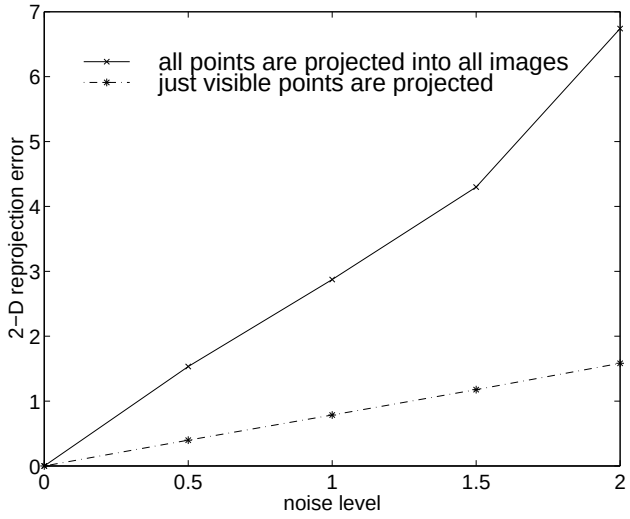

Figure 3: 2–D mean absolute pixel error of reprojections. These results were obtained for a window size of 20 frames with 5 overlapping frames between neighboring windows.

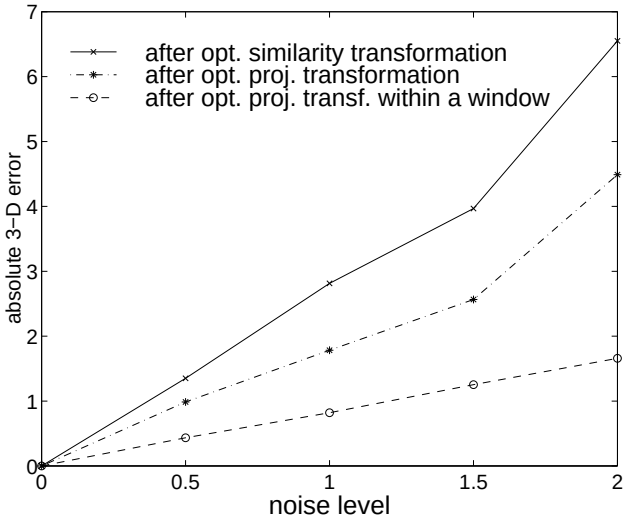

Figure 4: Mean absolute 3–D error for the same configuration in figure 3. It describes the mean difference between the true scene points and their corresponding reconstructed points after a specified transformation.

is used to create this distribution out from a uniform distribution within the range $0 \ldots 1$: $t(x) = \log(1-x)/\log(1-p)$. We also made the assumption that a fixed number of points must be seen in each frame (in our example 400). This means that for each lost point a new one is introduced. Figure 2 shows an example of the simulated visibility in the case of $p = 0.1$.

Reprojection error. One measure for the evaluation of errors is the distance between the projections of the reconstructed points into the reconstructed cameras and the original projections. Figure 3 shows the mean absolute error in the reprojections measured in pixels. It can be seen that for each reconstructed camera, the reprojected image coordinates of those points which are visible in this camera have an error in the magnitude of the noise (dotted curve). If even those points are projected into a particular camera which are just visible in other cameras, the error increases (solid curve). This increasing error is caused by errors during merging together the reconstructions of neighboring windows.

3–D error of the reconstruction. Another accuracy measure is the 3–D error of the reconstructed points. Since the reconstruction is done just up to an unknown similarity transformation, we have to determine that transformation which minimizes the 3–D error between the transformed reconstructed points and the true scene points.

Figure 4 shows the absolute 3–D errors in units of the simulated 3–D structure after that optimal similarity transformation (solid curve). The errors after transforming with an optimal projective transformation are less. This indicates a projective skew in the reconstruction. By applying a non–linear refinement of the self–calibration step, this skew should be reduced. If the optimal projective error is determined separately within each window, the 3–D error is even less. As in the 2–D case, this again indicates errors during the merging step.

Choice of the number of overlapping frames. As described in section 5, neighboring windows may be chosen to overlap by some frames to increase stability. Figure 5 shows the 3–D error after the optimal similarity transformation in dependence on the number of overlapping frames. It can be seen that a significant improvement can be noticed just for higher noise levels.

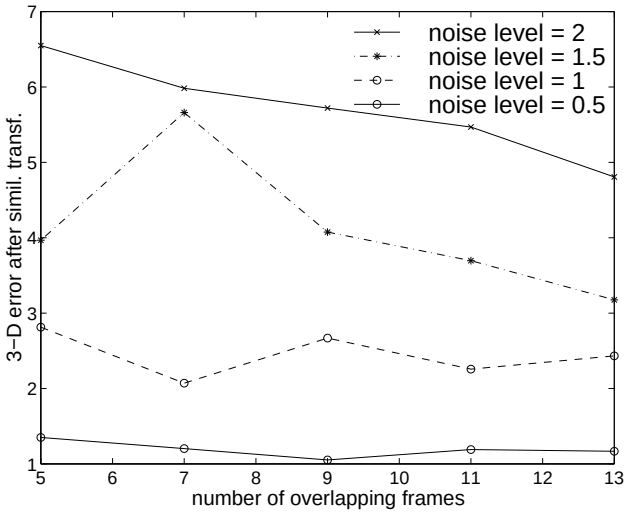

Figure 5: Error in dependence on the number of overlapping frames for different noise levels.

8 Conclusion

In this contribution we have shown how to apply the factorization method of Sturm and Triggs to real image sequences. We have seen that it is necessary to merge together partial solutions within a Euclidean framework which can be recovered by applying a self–calibration method. The experiments show that our method performs well, but improvements could be achieved by enhancing the merging step.

Acknowledgements. We would like to thank Marc Pollefeys from ESAT–PSI of the K. U. Leuven for giving us hints how to perform the self–calibration step.

References

[1] P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model aquisition from extended image sequences. In *Proceedings ECCV*, pages 683–695, 1996.

[2] O. Faugeras and L. Robert. What can two images tell us about a third one? Technical report, INRIA, Sophia Antipolis, 1993. technical report.

[3] R. I. Hartley. In defense of the eight–point algorithm. *PAMI*, 19(6):580–593, 1997.

[4] R. Koch, M. Pollefeys, and Luc Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proceedings ECCV*, pages 55–71. Springer, 1998.

[5] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings SIGGRAPH*, pages 31–45, 1996.

[6] Q.-T. Luong, R. Deriche, and O. Faugeras. On determining the fundamental matrix: Analysis of different methods and experimental results. Technical report, INRIA, Sophia Antipolis, 1993. technical report.

[7] R. Mohr and B. Triggs. Projective geometry for image analysis. In *Int. Symp. Photogrammetry and Remote Sensing*, July 1996. Tutorial.

[8] M. Pollefeys, R. Koch, and L. Van Gool. Self–calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings ICCV*, pages 90–95, Bombay, 1998.

[9] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C – The Art of Scientific Computing*. Cambridge University Press, New York, 1990.

[10] J. Shi and C. Tomasi. Good features to track. In *Proceedings CVPR*, pages 593–600. IEEE Computer Society Press, June 1994.

[11] P. Sturm and B. Triggs. A factorization based algorithm for multi–image projective structure from motion. In *Proceedings ECCV*, pages 709–720. Springer, 1996.

[12] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–153, 1992.

[13] P. H. S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–605, 1997.

[14] B. Triggs. Autocalibration and the absolute quadric. In *Proceedings CVPR*, pages 609–614, June 1997.