Accepted for publication in IJCV, early 2006

Affine Approximation for Direct Batch Recovery of Euclidian Structure and Motion from Sparse Data

Nicolas Guilbert[1], Adrien Bartoli[2], Anders Heyden[3]

[1]Centre for Mathematical Sciences
Lund University, Sweden

[2]LASMEA (CNRS / Université Blaise Pascal)
Clermont-Ferrand, France

[3]Division of Mathematics
Malmö University, Sweden

Abstract

We present a batch method for recovering Euclidian camera motion from sparse image data. The main purpose of the algorithm is to recover the motion parameters using as much of the available information and as few computational steps as possible. The algorithm thus places itself in the gap between factorisation schemes, which make use of all available information in the initial recovery step, and sequential approaches which are able to handle sparseness in the image data. Euclidian camera matrices are approximated via the affine camera model, thus making the recovery direct in the sense that no intermediate projective reconstruction is made. Using a little known closure constraint, the $\mathbf{F}_A$-closure, we are able to formulate the camera coefficients linearly in the entries of the affine fundamental matrices. The novelty of the presented work is twofold: Firstly the presented formulation allows for a particularly good conditioning of the estimation of the initial motion parameters but also for an unprecedented diversity in the choice of possible regularisation terms. Secondly, the new autocalibration scheme presented here is in practice guaranteed to yield a Least Squares Estimate of the calibration parameters.

As a bi-product, the affine camera model is rehabilitated as a useful model for most cameras and scene configurations, e.g. wide angle lenses observing a scene at close range. Experiments on real and synthetic data demonstrate the ability to reconstruct scenes which are very problematic for previous structure from motion techniques due to local ambiguities and error accumulation.

Keywords: Structure from motion, batch recovery, closure constraints, affine camera model, autocalibration, contraction mapping.

1 Introduction

The structure from motion problem has been studied extensively since the publication of the seminal paper [17] in 1981. The following efforts have been diverse and fruitful and a variety of algorithms now exist to recover the presumed camera motion and the observed 3D structure. Existing algorithms can be classified in two families, namely *batch* algorithms [16, 20, 4, 1, 14], which recover all pose and structure parameters in a single global step, and *sequential* algorithms [2, 19] where the parameters are recovered progressively as new views become available.

Also, substantial effort has been put into so-called autocalibration [5, 11, 13, 21], where the initially unknown

intrinsic parameters of the camera are recovered together with the pose.

In theory, the batch approaches should be the most suitable for off-line processing, where the data acquisition is concluded prior to processing, since all the available information is included in the initial estimation of the parameters. Also in theory, the sequential approaches should be relevant primarily for realtime applications such as navigation or interactive applications and should be avoided otherwise, since accumulation of the error is unavoidable. However, existing batch algorithms suffer from one major drawback, which is that roughly speaking all features have to be present in all images for the initial reconstruction to be feasible. Consequently, they are *per se* unapplicable to the common case where the images in a sequence have been taken from very different points of view and thus in majority have no features in common.

Certainly, successful suggestions have been made to extend the common batch approaches, i.e. factorisation schemes to the *missing data* case [14, 18, 1], however we here wish to distinguish that case from the *sparse data* case that we will be treating in this paper. There is a radical difference between 'patching' the measurement matrix for a small amount of missing features, and solving the problem for a measurement matrix that is say 90-99% unavailable.

The method presented here is in essence founded on the so-called closure constraints [25, 15], which we extend with a new variant, the $\mathbf{F}_A$-closure for the affine camera, based on a formulation originally developed in [27, 26]. This constraint allows a formulation of the camera matrix coefficients which is linear in the coefficients of the fundamental matrices.We can thus recover all the camera poses in a single computational step. This initial pose estimate subsequently needs refining which is done with a Euclidian bundle adjustment.

The work is thus has some similarities with [15] but differs on two important points: Firstly, the derivation and the final formulation of the $\mathbf{F}_A$-closure is of a surprising simplicity. Secondly, all the affine camera parameters are estimated, in particular the coordinates of the camera center. This eliminates the need for relative coordinates and thus the need for at least one point to be visible in all the frames, making the algorithm significantly more general.

Another contribution of the present work is a new autocalibration algorithm based on a *Contraction Mapping*. Assuming zero skew and unit aspect ratio, the algorithm in practice guarantees a least squares error on the estimated intrinsic parameters. The optimality of the algorithm is difficult (if possible at all to) prove theoretically. A tentative proof, based on several conjectures, has nevertheless been included in the appendix in order to strengthen the conclusion that the point of convergence is indeed optimal.

Although the affine camera model is often viewed as too simple for many if not most applications, our experiments show that within our framework, the affine approximation is clearly sufficient. The somewhat widespread rule of thumb that the depth of the observed object should be no more than 10% of the distance to the object is seriously undermined in our experimental section. The benefits of the presented algorithm are the following:

- All the available information on the epipolar geometry is used in the initialisation step. (Equation (14)).
- The reconstruction is direct in the sense that no intermediate projective reconstruction is done, and the hard part of autocalibration, i.e. detecting the plane at infinity, is performed inherently.
- Constraints related to equality/proximity of cameras are easily included. (Section 4.2).
- Constraints imposed by some cameras being known, even only approximately, are naturally included. (Section 4.2).
- Constraints related to equality of 3D points are naturally accounted for.
- Constraints modelling smoothness of the camera trajectory are easily included. (Section 4.1).
- Local ambiguities may in general be overcome, since the problem is solved globally. (Section 6).
- The algorithm is very fast, with an execution time which is linear in the number of estimated cameras for certain common types of scene configurations. (Sections 3 and 6).

Summing up, the presented algorithm is very robust, in particular to local ambiguities. Generally speaking, this is the consequence of the choice of the affine camera

model, which is simple and thus potentially leading to a good conditioning of the problem, and yet sufficiently rich to adequately model the problem at hand. One specific advantage of the presented approach is the ability to naturally handle a closed sequence, i.e. a sequence in which the same 3D features appear on several occasions in the sequence. Another advantage over factorisation algorithms is robustness towards outliers, since the algorithm is based on fundamental matrix calculations, which can be done robustly.

The paper is organised as follows: The notation and the elementary background is given in Section 2. In Section 3 the $\mathbf{F}_A$ constraint is introduced together with details on how to use it. Different types of regularisation of the problem are described in Section 4. The autocalibration is described in Section 5 and the appendix. Experiments on real and synthetic data are presented in Section 6, the error analysed in Section 7 and conclusions given in Section 8.

2 Notation and Background

2.1 Generalities

We denote 2D homogeneous image points by a subscripted $\mathbf{x}$, 3D homogeneous points by $\mathbf{X} = [\bar{\mathbf{X}} \quad 1]^\top$ and 3×4 camera matrices by $\mathbf{P}$. These entities are related by the following *projection*:

$$\lambda_{ij}\mathbf{x}_{ij} = \mathbf{P}_i\mathbf{X}_j \quad i = 1 \dots m,\ j = 1 \dots n, \tag{1}$$

where m and n denote the number of views and 3D object points respectively, and λ is the *projective depth* of the given point.

The fundamental matrix $\mathbf{F}_{21}$ encapsulates the geometrical relationship between two views (for simplicity of notation we consider views 1 and 2). It constrains the position of corresponding image points $\mathbf{x}_{1j}$ and $\mathbf{x}_{2j}$ through the relation

$$\mathbf{x}_{1j}^\top \mathbf{F}_{21}\mathbf{x}_{2j} = 0, \quad j = 1 \dots n. \tag{2}$$

The line defined by $\mathbf{F}_{21}\mathbf{x}_{2j}$ is the epipolar line of $\mathbf{x}_{2j}$ in image 1, i.e. the line joining the epipole $\mathbf{e}_{12}$ and $\mathbf{x}_{1j}$.

We now introduce the perspective camera matrix, which models a classical pinhole camera. It may be decomposed as

$$\mathbf{P} = \underbrace{\begin{bmatrix} \gamma_x f & sf & x_c \\ 0 & \gamma_y f & y_c \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} \mathbf{r}_1^\top & t_1 \\ \mathbf{r}_2^\top & t_2 \\ \mathbf{r}_3^\top & t_3 \end{bmatrix}}_{[\mathbf{R}^\top \quad \mathbf{t}]} \tag{3}$$

$$\tag{4}$$

$$= \mathbf{K}\,\mathbf{R}^\top[\mathbf{I}_{3\times 3} \quad -\mathbf{c}], \tag{5}$$

where f denotes the focal length, γ_x and γ_y the scaling along the images' x- and y-axes i.e. $a = \frac{\gamma_y}{\gamma_x}$ is the aspect ratio. s is the skew and (x_c, y_c) the principal point. The $\mathbf{r}_i$ are the three columns of a rotation matrix $\mathbf{R}$ and $\mathbf{c} = -\mathbf{R}^\top(t_1, t_2, t_3)^\top$ is the camera centre. In general, x_c and y_c are highly correlated to the camera rotation, highly ambiguous and only of interest if the rotation needs to be determined precisely. They are thus often, and will be in the sequel be, assumed to be zero. γ_x and γ_y are in general not written out explicitly, since they algebraically yield an overparameterisation of the intrinsic calibration, i.e. in the literature $\gamma_x \equiv 1$ and (3) becomes

$$\mathbf{P} = \begin{bmatrix} f & sf & x_c \\ 0 & af & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^\top & t_1 \\ \mathbf{r}_2^\top & t_2 \\ \mathbf{r}_3^\top & t_3 \end{bmatrix}. \tag{6}$$

Formulation (3) will be needed to understand the relationship between the perspective and affine image formation processes. The affine camera matrix has the form

$$\mathbf{P}' = \begin{bmatrix} \bar{\mathbf{P}} & \mathbf{t} \\ \mathbf{0}_3^\top & 1 \end{bmatrix}, \tag{7}$$

where the uncalibrated state is expressed by the prime ($'$). When calibrated it may be decomposed as

$$\mathbf{P}_A = \begin{bmatrix} \hat{\gamma_x} & s & 0 \\ 0 & \hat{\gamma_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^\top & t_1 \\ \mathbf{r}_2^\top & t_2 \\ \mathbf{0}_3^\top & 1 \end{bmatrix}, \tag{8}$$

where $\hat{\gamma_x} = \gamma_x$ and $\hat{\gamma_y} = \gamma_y$ are the scaling factors from (3). The calibrated affine camera model is an approximation to the calibrated perspective model where the focal length is considered infinite and the camera centre has been retracted to infinity, i.e.

$$\mathbf{P}_A = \lim_{\nu\to\infty} \begin{bmatrix} \nu\gamma_x f & \nu sf & x_c \\ 0 & \nu\gamma_y f & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^\top & t_1 \\ \mathbf{r}_2^\top & t_2 \\ \mathbf{r}_3^\top & t_3 + (\nu-1)f \end{bmatrix} \frac{1}{\nu f}. \tag{9}$$

where ν indicates how much the camera has retracted from the scene, measured in multiples of the focal length. In Figure 1 the process is illustrated. For $\nu = 1$, the camera matrix is the one from (3). As ν increases, the camera centre retracts in the negative direction of the z-axis and the focal length, i.e. the distance between the camera centre and the image plane, increases accordingly.

The disappearance of $\mathbf{r}_3^\top$ in (8) implies that the projections onto the image plane are parallel. Also, in the affine case, the projective depths $\lambda_{ij} = 1$.

The affine fundamental matrix has five non-zero entries defined up to scale, i.e. four degrees of freedom:

$$\mathbf{F}_{21} = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{bmatrix}. \tag{10}$$

Computing the affine fundamental matrix thus requires at least four point correspondences. However as many points as possible should be included to minimise the effect of noise. A closed-form solution for the Maximum Likelihood Estimate of the affine fundamental matrix exists [12].

3 A Simple Formulation of the Affine Closure Constraints

In [25], Triggs introduces the so-called $\mathbf{F} - \mathbf{e}$ closure constraint, namely $\mathbf{F}_{21}\mathbf{P}_2 + [\mathbf{e}_{21}]_\times \mathbf{P}_1 = \mathbf{0}$ and the $\mathbf{e} - \mathbf{G} - \mathbf{e}$ closure. In [15] closure constraints for the affine camera model are derived. We will here give a simple derivation of an alternative closure constraint which is specific to the affine case, the $\mathbf{F}_A$-closure. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ denote the projections of $\mathbf{X} \in \mathbb{P}^3$ onto two images. Combining equations (2) and (1) we obtain:

$$\mathbf{X}^T \underbrace{\mathbf{P}_1^T \mathbf{F}_{21} \mathbf{P}_2}_{\mathcal{S}} \mathbf{X} = 0, \quad \forall \mathbf{X} \in \mathbb{P}^3, \tag{11}$$

which is a quadratic form. Consequently $\mathcal{S}$ is skew symmetric, i.e. $\mathcal{S}^T = -\mathcal{S}$ as noted in [6].

In the affine case, the structure of $\mathcal{S}$ becomes particularly simple, which stems from the structure of the camera matrices (7) and the affine fundamental matrix (10):

$$\mathcal{S} = \mathbf{P}_2^T \mathbf{F}_{21} \mathbf{P}_1 = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{a} \\ -\mathbf{a}^T & 0 \end{bmatrix}. \tag{12}$$

The upper left $\mathbf{0}_{3\times 3}$ matrix is the result of the corresponding bilinear term containing either a zero-coefficient either from the fundamental matrix or one of the camera matrices. The rest of the structure of $\mathcal{S}$ is a consequence of the skew-symmetry, and *concerns only linear terms* in the entries of the camera matrices since these terms include the lower right 1 of either $\mathbf{P}_1$ or $\mathbf{P}_2$. By rearranging the equations in (12) these turn into four linear constraints on the coefficients of $\mathbf{P}_1$ and $\mathbf{P}_2$:

$$\begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} \bar{\mathbf{P}}_1 & \mathbf{t}_1 \\ \bar{\mathbf{P}}_2 & \mathbf{t}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0}_3 & -e \end{bmatrix}}_{\mathbf{r}_{12}}, \tag{13}$$

where a, b, c, d and e are entries of $\mathbf{F}_{21}$ in (10). The above formulation was originally developed in [26]. The constraints of (13) apply for each pair of views $\{\mathbf{P}_{i_1}, \mathbf{P}_{i_2}\}$, $i_1 \neq i_2$, provided $\mathbf{F}_{i_1 i_2}$ is defined. Affine trifocal or quadrifocal tensors could be used as well by extracting fundamental matrices, or along the lines of [15]. We construct a linear system of equations using (13) as the building block, with the form $\mathbf{S}\mathcal{P} = \mathcal{R}$:

$$\begin{bmatrix} & & \\ & \mathbf{s}_{12} & \\ & \mathbf{s}_{1 i_1} & \\ & \vdots & \\ & \mathbf{s}_{i_k i_m} & \end{bmatrix} \begin{bmatrix} \bar{\mathbf{P}}_1 & \mathbf{t}_1 \\ \bar{\mathbf{P}}_2 & \mathbf{t}_2 \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \bar{\mathbf{P}}_m & \mathbf{t}_m \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{12} \\ \mathbf{r}_{1 i_1} \\ \vdots \\ \mathbf{r}_{i_k i_m} \end{bmatrix}, \tag{14}$$

where $\mathbf{r}_{i_1 i_2}$ is the right hand side in equation (13) and $\mathbf{s}_{i_1 i_2}$ are $1 \times 2m$ row vectors

$$\mathbf{s}_{i_1 i_2} = [\ldots \quad \underbrace{a \quad b}_{\text{First block}} \quad \ldots \quad \underbrace{c \quad d}_{\text{Second block}} \quad \ldots], \tag{15}$$

dots indicating an adequate number of zeros. One possible structure for $\mathbf{S}$ is shown in Figure 2. Since the global 12-parameter affine coordinate system has not been specified prior to solving the system, the solution is a 12-dimensional subspace which may be recovered by SVD. However, this is rather inefficient compared to using sparse linear solvers, which require the system to be of full rank. Full rank is obtained by choosing an overall affine coordinate system arbitrarily. This can be done e.g. by fixing all the coefficients of a camera $\mathbf{P}_a$ and 4 coefficients of some arbitrary second camera $\mathbf{P}_b$ and modifying

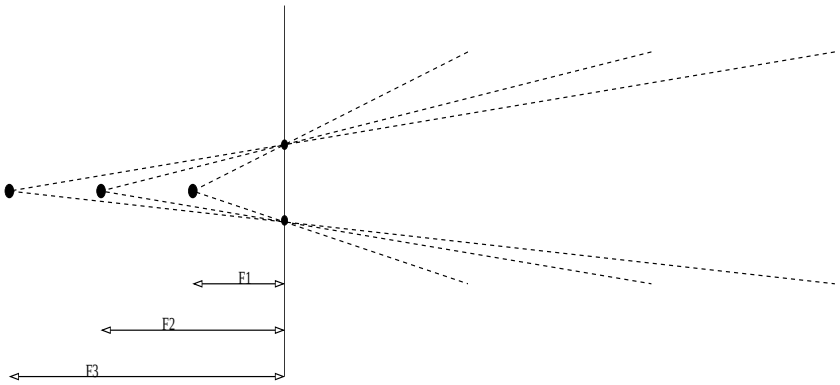


Figure 1: **From perspective to affine projection.** As the camera centre away retracts from the image plane, the focal length increases and the projection becomes more and more parallel. The figure shows the situation for three different focal lengthes f_1, $f_2 = 2f_1$ and $f_3 = 3f_1$.

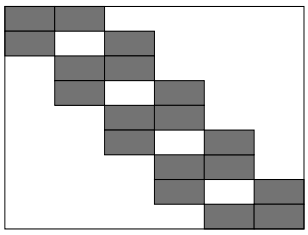


Figure 2: **Structure of the central design matrix** constraining the camera matrices. The structure shown is for a minimal configuration, i.e. $2m-3$ block-row entries for m views, before fixing the 12-parameter gauge freedom (see text for further details).

the design matrix accordingly. The choice of affine reference frame is important. An example of an a priori valid but bad choice would be the matrices

$$\mathbf{P}_a = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{P}_b = \begin{bmatrix} 0 & 0 & 0 & 0 \\ * & * & * & * \end{bmatrix},$$

since this would force all the other cameras to be of the form

$$\begin{bmatrix} 0 & 0 & 0 & * \\ 0 & 0 & 0 & * \end{bmatrix}.$$

A choice of frame that seems to work well (the one used in the experimental section) is given by the pair

$$\mathbf{P}_a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{P}_b = \begin{bmatrix} 0 & 1 & 1 & 0 \\ * & * & * & * \end{bmatrix}.$$

The design matrix will often have a diagonal structure, since this amounts to each 3D point being visible in a limited number of consecutive frames. Thus the advised solver would be based on band or profile Cholesky factorisation with back-substitution. The execution time of the band Cholesky being linear in the height of the matrix, the presented algorithm becomes linear in the number of recovered camera matrices.

Alternatively, a supernodal approach could be used in case the sparseness structure is more *ad hoc*. For a more in-depth description of the suggested solvers, see [9] and e.g. [3] for implementational details.

4 Regularising Camera Motion

One well-known and common problem in structure from motion is ill-conditioning, and the ability to regularise the problem using prior knowledge thus becomes particularly important.

Within this framework, different types of prior knowledge of the motion parameters may be formulated as linear constraints among the parameters, which may be naturally incorporated into the design matrix $\mathbf{S}$ in (14). The two types of constraints we consider are smoothness constraints on the trajectory and equality of some cameras.

4.1 Smoothness of the Trajectory

The system (14) is in general over-determined unless a minimal case is being dealt with. Thus, we will in general consider solving (14) in a minimal least squares sense through the normal equations:

$$\mathbf{S}^\top \mathbf{S}\mathcal{P} = \mathbf{S}^\top \mathcal{R}. \tag{16}$$

One straightforward way of imposing soft constraints would thus be to augment $\mathbf{S}$ with the desired linear constraints. For instance, smoothness of the camera trajectory and orientation could be obtained by softly imposing equality between adjacent cameras in the sequence. The linear equality constraints would be $\mathbf{S}_s\mathcal{P} = 0$ with

$$\mathbf{S}_s = \begin{bmatrix} 1 & 0 & -1 & \dots & & & & \\ 0 & 1 & 0 & -1 & \dots & & & \\ & & & \ddots & \ddots & & & \\ & & & \dots & 1 & 0 & -1 & 0 \\ & & & & \dots & 1 & 0 & -1 \end{bmatrix} \tag{17}$$

and may be included in (16) as

$$(\mathbf{S}^\top \mathbf{S} + \alpha \mathbf{S}_s^\top \mathbf{S}_s)\mathcal{P} = \mathbf{S}^\top \mathcal{R}, \tag{18}$$

where α expresses the desired strength of the constraint. Note that imposing the constraints is likely not to affect the execution time of an implementation based on sparse diagonal solvers. If we let k denote the number of views in which a minimum number of common features appears, i.e. the number of consecutive views "sharing" a fundamental matrix, the structure of $\mathbf{S}^\top \mathbf{S}$ is essentially $(2k-1)$-diagonal. k being of an order of magnitude 10, the structure won't be affected by the pentadiagonal structure of $\mathbf{S}_s^\top \mathbf{S}_s$.

4.2 Equality of Cameras and Fixed Cameras

As described in (15) each camera corresponds to a pair of columns in the design matrix. Enforcing equality among two cameras simply amounts to adding the corresponding columns. In contrast to the smoothness constraints, the camera equality constraint is not imposed as an additional term appearing in the minimal least squares solution to (14), but as a hard constraint that is inherently satisfied.

If a given camera has a known position, this is easily imposed by multiplying the corresponding columns in $\mathbf{S}$ by the camera matrix and substracting the result on the right hand side of (14).

5 Recovery of Euclidian Motion and Structure

Recovery of the Euclidian perspective equivalents to the affine camera matrices is done in two steps, a separation of the affine intrinsic and extrinsic parameters, i.e. a calibration step, followed by a an upgrading from affine to full perspective cameras.

5.1 Euclidian Calibration

Autocalibration for affine cameras is classically done as described in the method in [22], i.e. by assuming skew $s = 0$ and aspect ratio $\frac{\hat{\gamma}_x}{\hat{\gamma}_y} = 1$ and determining an upgrading affine transformation $\mathbf{H}$ that maps (7) to (8), i.e. $\mathbf{P} = \mathbf{P}\mathbf{H}$.

We here propose an alternative iterative scheme which in practice guarantees convergence to a unique Least Squares Estimate of the three parameters up to an overall scaling.

The basic idea of the algorithm is to iteratively find an affine transformation $\mathbf{H}_c$ that will transform the uncalibrated affine cameras so as to get them as close to the calibrated form (8) as possible. The idea is very simple, although the formal presentation within the framework of the contraction mapping theorem tends to make it somewhat opaque. The detailed presentation has been included in an appendix in order to strengthen the understanding of the algorithm.

We begin by considering that the information needed for the calibration of an affine camera is contained in 1) the upper triangular form of the intrinsic calibration matrix $\mathbf{K}$, and 2) in the orthonormality of the two vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ defining the rotation of the camera. In order to recover the form (8) we start out by aligning our existing estimate of the form (8) with a plausible candidate, the plausible candidate being the current $\mathbf{r}_1$ and $\mathbf{r}_2$ of each camera. These current $\mathbf{r}_1$ and $\mathbf{r}_2$ are obtained via QR-factorisation of each of the current matrices and subsequently stacked in a large matrix. In this alignment, the aligned estimate will hopefully inherit some of the structure of the target. Furthermore, repeating the alignment brings the camera matrices as close as possible to the desired form (see appendix). The algorithm proceeds as follows:

1. Stack the 2×3 camera matrices in a $2m \times 3$ matrix $\mathcal{P}$;
2. Perform a QR factorisation of each of the camera matrices;
3. Stack the resulting rotation matrices in a $2m \times 3$ matrix $\mathcal{R}$.
4. Align $\mathcal{P}$ to $\mathcal{R}$, i.e. minimise/reduce the Frobenius norm $\|\mathcal{P}\mathbf{H} - \mathcal{R}\|_F$.
5. Go to 1. unless the algorithm has converged.

Even though the algorithm is iterative, it converges very fast, 1-3 iterattions tend to suffice to obtain valid parameters, and each iteration takes a few milliseconds ($m > 100$ cameras).

In the appendix, it is shown that given that the algorithm converges, the point of convergence is the Least Squares Estimate of the calibration parameters. However, convergence can presently not be guaranteed formally, only experimentally (see Figure 16).

5.2 Affine-to-Perspective Upgrading

Once the form (8) has been recovered, the perspective equivalents (3) are obtained by taking advantage of the $f \leftrightarrow \gamma_x$ ambiguity: Since (3) is overparameterised with respect to f and γ_x, we are free to initially assume $\gamma_x f = \hat{\gamma_x}$ and subsequently $\gamma_x = 1$.

Although $\mathbf{r}_3$ has disappeared in the limit (9), it may be easily recovered since $\mathbf{R}$ is a rotation matrix with $\det(\mathbf{R}) = +1$. To summarise, the Euclidian matrices in the form (6) are recovered by

$$\begin{cases} \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \\ f = \hat{\gamma_x} \\ a = \frac{\hat{\gamma_y}}{\hat{\gamma_x}} \end{cases} . \tag{19}$$

6 Experiments

6.1 Synthetic Data

Two experiments on synthetic data are presented. The first concerns the reconstruction of a large cubic point cloud from circular motion and the second the reconstruction of a room seen from within, also from a circular camera trajectory.

Object Centered Trajectory

The first dataset, the cubic point cloud, is shown in Figure 3 (Top). It consists of 300 3D points evenly distributed in the cube $[0, 5] \times [0, 5] \times [0, 5]$ and of 30 cameras with focal length $f = 100$ distance units equidistantly placed on a circular path centered at $(0, 0, 0)$. Each frame contains features which are visible in the 9 following frames. Gaussian noise with $\sigma = 1$ is present in the images. Figure 3 (Center) shows the initial reconstruction of the camera trajectory with the method from Section 3 and 5 and in Figure 3 (Bottom) an alternative reconstruction where equality has been assumed between the first and the last camera (closed the sequence). Clearly, the initial reconstructions capture the overall structure of the scene and the motion, thus allowing for the subsequent bundle adjustment to converge to the global minimum. One point of special interest (see the discussion in Section 5.2) is the fact that within this framework, the affine camera model approximates the perspecitive camera sufficiently well, even though the depth of the object is approximately the same as the distance to the object, i.e. a lot more than the 10% that are usually considered the upper limit.

In this experiment (Figure 3,Bottom) equality was assumed between the first and the last camera. It should be noted that in general, known relative positions between any of the cameras can be imposed. This would be done

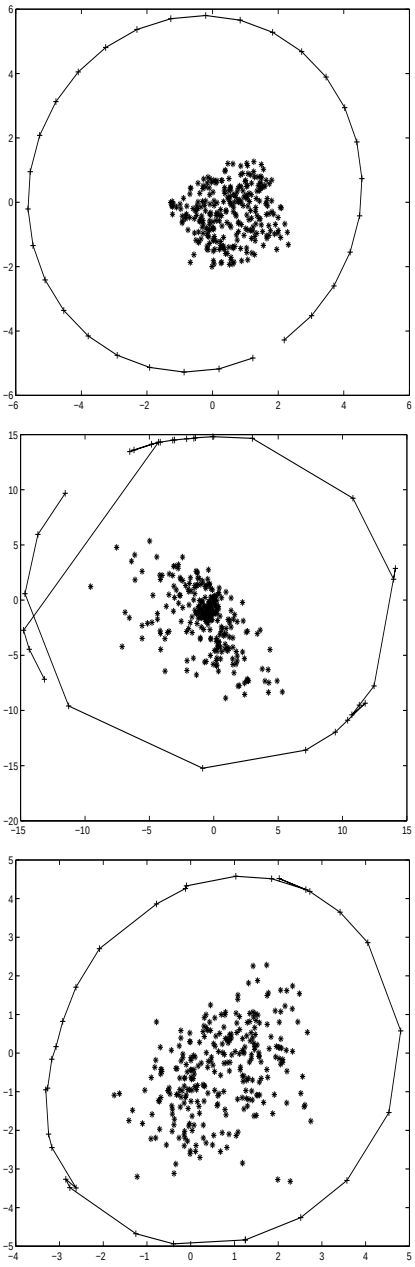


Figure 3: **Reconstruction, object centered configuration (closed sequence)** Top: original configuration. Center: initial reconstruction using affine approximation. Bottom: same as (b) but assuming equality between the first and last camera.

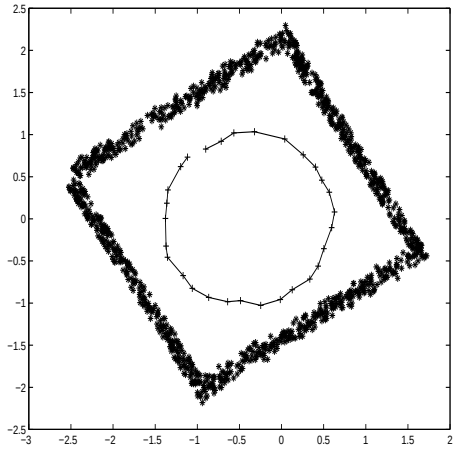


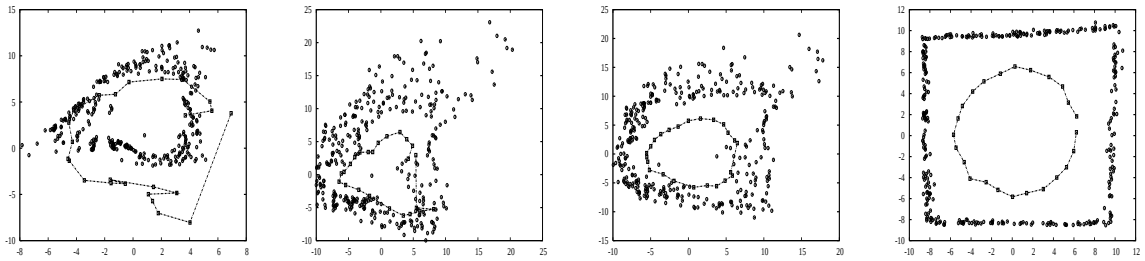
Figure 4: **Reconstruction of a structure observed from within:** experimental setup. The camera describes a circular trajectory and has an angular aperture of $\frac{\pi}{3} < \alpha < \frac{\pi}{2}$.

by eliminating the parameters of all but one of the cameras of such a set of mutually fixed cameras from the left hand side of (14), as discussed in Section 4.

Interior of a Room

The second experiment on synthetic data describes a scene configuration that is known to be quite difficult, i.e. the case of a camera taking views of the interior of a room in order to reconstruct the walls and ceiling, see Figure 4: Each feature is present only in a little subset of the frames, planar structures dominate the scene thus leading to focal length $\leftrightarrow$ translation ambiguities and finally if the sequence is reconstructed sequentially, the accumulated error will be likely to be irrepairable. The experimental setup is shown in Figure 4.

It consists of a square 'room' with 600 points evenly distributed on each of the four walls and the roof, the 2D image points being subject to Gaussian noise with $\sigma = 1$. The camera describes an almost circular path within the room, pointing inwards. The initial reconstruction shown in Figure 5 (Top) and Figure 6 (a) clearly captures the overall closed structure of the scene and a subsequent bundle adjustment converges to the global min-



(a) (b) (c) (d)

Figure 6: **Post-processing with bundle adjustment.** After a few bundle adjustment steps (respectively 0, 2, 4,and 6) the initial reconstruction has gotten very close to the desired result.

imum. However, better results can be obtained by imposing equality of the first and the last camera (Figure 5, Center) and smoothness of the camera centres trajectory (Figure 5, Bottom).

Note that some of the cameras have been reconstructed outside the room. This is however only a problem if the 3D points they see were to be reconstructed behind the camera, which can be avoided by retracting the camera / assigning a higher focal length.

6.2 Real Data

Two experiments were performed on real data, on the Dinosaur sequence and the Palazzo Pitti sequence.

The Dinosaur Sequence

The Dinosaur sequence consists of 37 images, of which the first and the last are known to coincide. A total of 1888 3D points were tracked across the sequence using the KLT-tracker [23]. Figure 7 shows a sample image from the sequence (Left) and the structure of the camera configuration (Right), i.e. whether a given pair of frames has sufficient overlap to allow for a fundamental matrix among them. Note that the first and the last images haven't been registered to each other, although they are practically identical. The reconstruction is done without imposing the constraint that the first and the last camera positions are equal, and the result is shown in Figure 8 (Top). The reprojection error after the initial affine reconstruction is 5.4 pixels, and the system (13) is solved in 2.1 milliseconds (Pentium IV@1.8Ghz). After Euclidian bundle adjustment, the final reprojection error is 0.62 pixels and the scene looks like Figure 8 (Bottom).

The dinosaur sequence is known to have been taken with a high focal length, implying that the data set would be particularly well adapted for an algorithm based on the affine camera model. In the next experiment, this is not the case.

Palazzo Pitti sequence

The second experiment has a sequence of 41 images as input data, of the type shown in Figure 9 (a). The images were taken within a room in the Palazzo Pitti in Florence with a wide angle camera. This type of sequence is particularly difficult to reconstruct, which is probably best illustrated by the absence of such reconstructed scenes using structure from motion in the literature. The difficulty lies in the fact that the measurement matrix is sparse, which makes the problem impracticable for factorisation schemes. Also, no feature appears in all the images, which is a problem for sequential approaches since the absence of a common reference allows the camera positions to drift, and the error accumulation becomes exceptionally severe. As it can be seen from Figure 9b, fundamental matrices exist between the first and the last images in the sequence. It has been 'stitched' together, which was done semiautomatically, i.e. a few points were selected in the first and the last frame and the rest of the matching was guided by the induced homography between the images. A total of 2564 point matches were reconstructed,

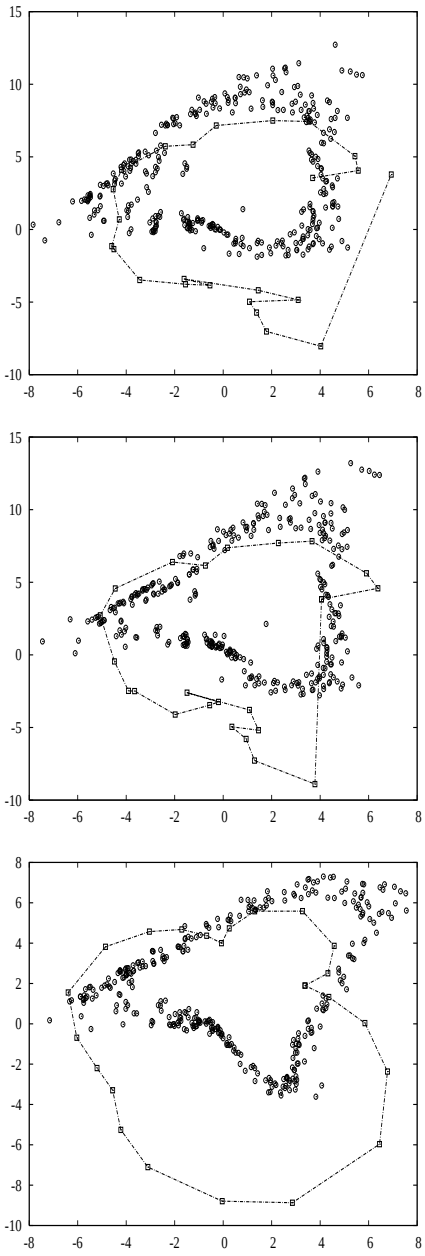


Figure 5: **Reconstruction of a structure observed from within** Top: Initial reconstruction. Center: initial reconstruction assuming equality between first and last camera. Bottom: assuming equality between the first and the last camera and with smoothening of the trajectory.

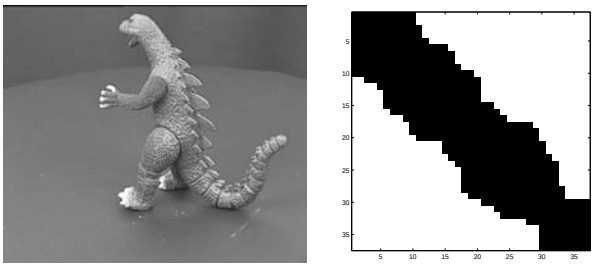


Figure 7: **The Dinosaur sequence** Left: sample image from the sequence. Right: matrix showing the availability of a fundamental matrix between two views.

as shown in Figure 10.

The experiment illustrates an important point, namely that imposing equality of reconstructed 3D points is done inherently. When applying a sequential approach, the constraints contained in this equality have to be imposed in a post-processing step, e.g. as described in [10]. Here, the 3D points never get reconstructed as different instances and the problem thus never occurs.

Note that for both experiments, Euclidian structure and motion were obtained directly, i.e. without an intermediate projective reconstruction. Autocalibration was done by applying a few (2-3) iterations.

7 Error Analysis

In order to assess the error on the reconstruction algorithm, the setup shown in Figure 11 is used as a test bench. m cameras are laid out on a circular path, all pointing inwards. The observed object consists of n 3D points uniformly distributed within a cube.

7.1 Comparison to Existing Methods

Recovery of Motion Parameters

The proposed algorithm is compared to two methods: The first is the classical factorisation from [24]. The second is the classical sequential schemes from [2] where 6 cameras and all the 3D points are computed from the first frames in

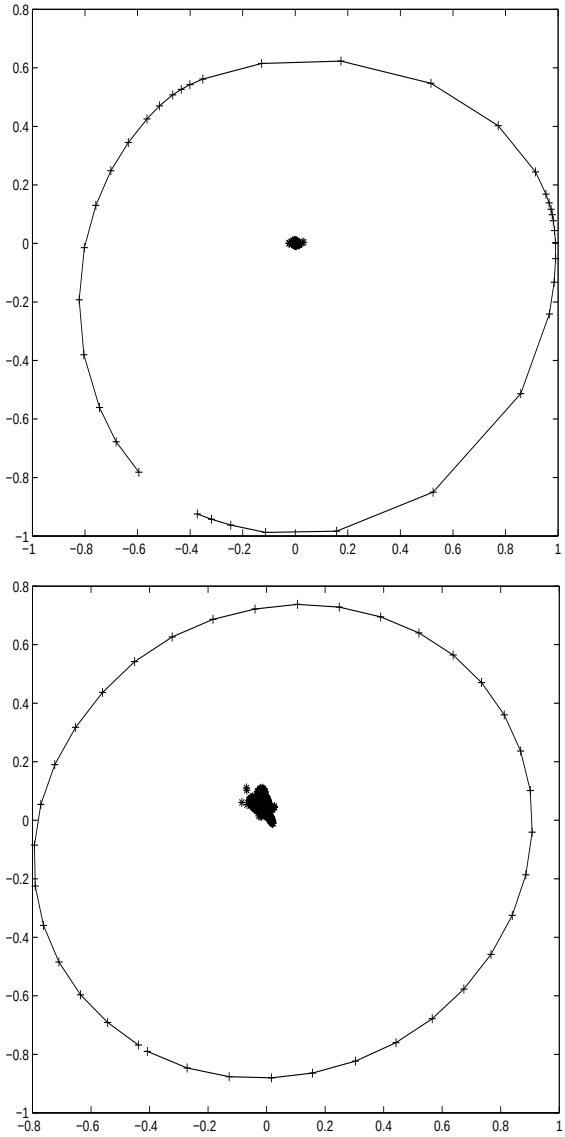


Figure 8: **Reconstruction from the Dinosaur sequence** Top: initial reconstruction (assuming proximity between the first and last camera positions) Bottom: final reconstruction after Euclidian bundle adjustment.

Figure 9: **The Palazzo Pitti in Florence** (a) sample image from the sequence (b) matrix showing the availability of a fundamental matrix between two views (a black entry signifies presence)

the sequence, followed by a series of resections and intersections in order to recover the remainder of the motion and structure [2]. In the experiment, $m = 12$ cameras and $n = 100$ 3D points were used, with Gaussian noise added to the x- and y-coordinates of the image points, all 3D points being visible in all views. As it can be seen from Figure 12, the presented batch algorithm lies between the resection/intersection-approach and classical factorisation. The error is the mean Euclidean distance between the measured and the reprojected points.

Autocalibration

In the traditional approach to affine autocalibration described by Quan in [22] the problem is formulated as that of solving a set of homogeneous quadratic equations in a least squares sense which is done using Levenberg-Marquardt minimisation. Such an approach is generally prone to stranding in local minima.

In the following experiment, a varying number (2,4,8,16 and 32) of random cameras were generated and transformed by a random 3×3 transformation $\mathbf{H}_r$.

The success rates of Quan's and the proposed algorithm were compared together with their execution times. Quan's algorithm reached the global minimum approximately 90% of the time for $m > 2$, compared to 100% for the contraction mapping scheme we propose. Also, the execution times were significantly lower for the contraction mapping scheme.

The results, success rates and execution times, are shown in Figure 13 (implementation on a standard PC).

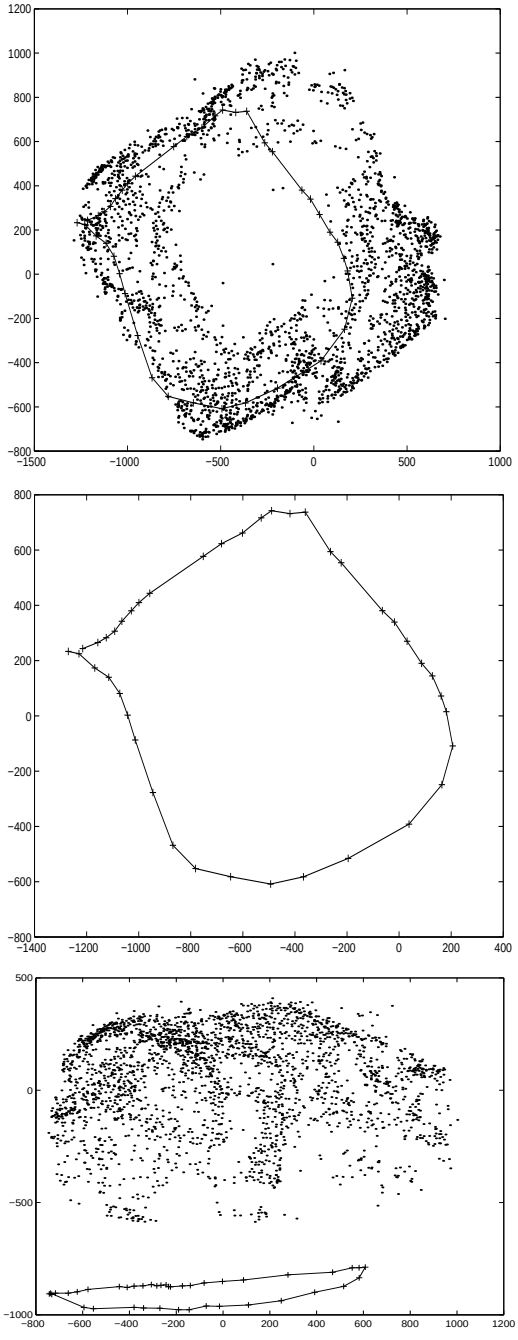


Figure 10: **Reconstruction of a room in the Palazzo Pitti** Top: structure, top view. Center: motion, top view. Bottom: structure, profile.

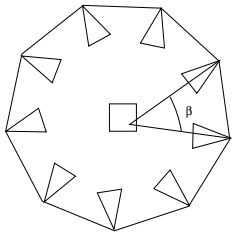


Figure 11: **The experimental setup for the error analysis**: m affine cameras pointed inwards on a circular path, observing a cubic cloud of n uniformly distributed points. β indicates the angle between the focal axis of two neighbouring cameras and is used as a measure of the baseline

.

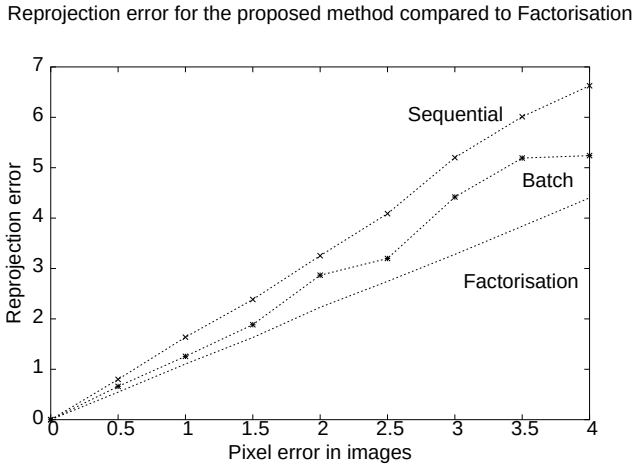


Figure 12: **Comparison** between the classical factorisation method from [24], the proposed batch method and a sequential scheme where an initial structure is computed using 6 cameras, whereupon the remaining cameras are obtained by resectioning. The graph shows the reprojection error as a function of the (Gaussian) image noise. The proposed batch algorithm performs better than the resection/intersection approach and is close to the factorisation algorithm.

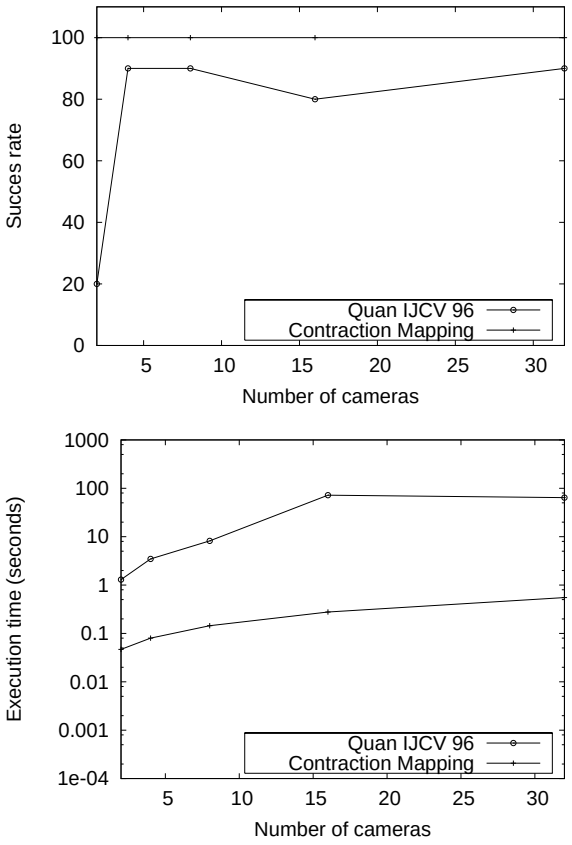


Figure 13: **Comparison to traditional affine calibration, see [22]** Top: Success rate (%). The existing state-of-the-art algorithm reaches the global minimum $\approx 90\%$ of the time. The proposed algorithm shows a 100% success rate. Bottom: Execution times. Even though the proposed algorithm is iterative, every iteration is very fast, only few iterations are needed and the execution time scales linearly with the number of cameras.

7.2 The Influence of Various Parameters

Width of the camera baseline

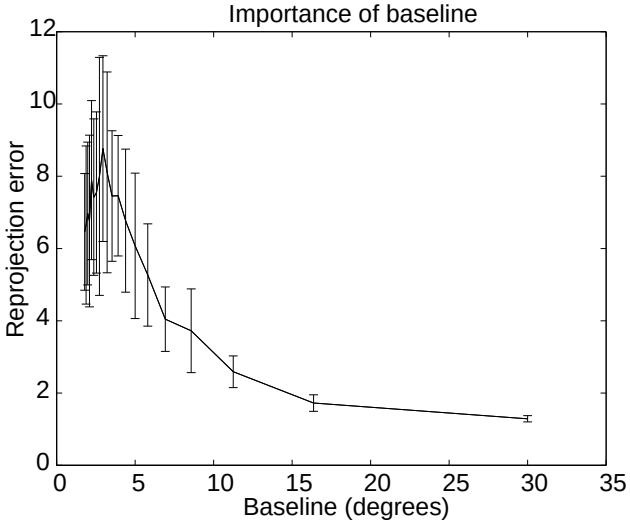
In order to assess the importance of the width of the baseline, the experiment shown in Figure 14a was performed. Again, $m = 12$ cameras were used, positioned on the arc of the circle of Figure 11 and the angle of incidence of their focal axis β varied from 2° to 30°, i.e. the overall baseline varied from 24° to 360°. The noise in the image is Gaussian noise $\sigma = 1$ pixel. Fundamental matrices were computed between each view and 10 of its neighbours. The reprojection error is seen to peak for $\beta \approx 7^\circ$. For lower values of β, the constraints imposed on the structure by the cameras are so loose that they are easily satisfied. For higher values, the computation of the fundamental matrices is getting well-conditioned, thus yielding lower reprojection errors.

Sensitivity to degenerate matching tensors

In this experiment ($m = 12$, $n = 20$, and $\beta = 18^\circ$, image noise $\sigma = 1$ pixel) the sensitivity of the algorithm with respect to the number of deficient fundamental matrices is investigated. This is done by successively setting the images $2 \dots m-1$ equal to the m'th, thus ensuring degenerate fundamental matrices among them. The result is shown in Figure 14b, where the reprojection error is plotted against the percentage of equal views in the sequence. As the number of equal views increase, i.e. the degeneracies become more numerous, the reprojection error is actually seen to decrease. When all the views are the same, the configuration is globally degenerated and the reprojection error is meaningless. Again, the more equal views in the sequence, the looser the constraints on the structure, eventually leading to a lower reprojection error. It is however noteworthy that degenerate fundamental matrices are not invalidating the system.

8 Conclusions

A batch algorithm for recovering the Euclidian camera motion from sparse data was presented. A new formulation of the closure constraint for the affine camera allowed for a formulation of the camera matrix coefficients which



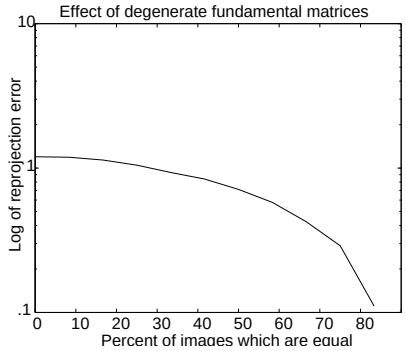


Figure 14: **The influence of various parameters**: **(a)** The effect of enlarging the baseline. The figure shows the decrease in error as the baseline gets larger (from $\beta = 2°$ to $\beta = 30°$). $m = 12$ views were used and $n = 150$ points, each camera was related to 10 neighbours using fundamental matrices. Image noise $\sigma = 1$ pixel. **(b)** The effect of degenerate camera matrices in the sequence. The graph shows the reprojection error as a function of the percentage of views which are equal in the sequence ($m = 12$, $n = 150$, $\beta = 18°$, $\sigma = 1$ pixel).

is linear in the fundamental matrix coefficients. Using the affine camera matrix as a model for the perspective camera, this allows approximated Euclidian camera matrices to be recovered via the solution of a single linear system followed by an affine-Euclidian calibration step. Several types of constraints are naturally included, e.g. proximity of given cameras or equality of 3D points.

A highly robust autocalibration scheme applied to affine-Euclidian calibration was introduced, a scheme that is generalisable to projective-Euclidian calibration.

Experiments on synthetic and real data showed that the algorithm performs well, i.e. succeeded in reconstructing even unprecedented difficult scenes.

As it is often the case for computer vision algorithms, a successful outcome is highly dependent on various implementational details. Hence a suggested implementation including a tracker, the presented algorithm and bundle adjustment is freely available for download at [8].

A On Euclidian Autocalibration

A.1 The Contraction Mapping Theorem

Before addressing the central problem, we define a *contraction* and introduce a common tool from functional analysis, the so-called *Contraction Mapping Theorem* here reproduced from [7]:

Definition A.1. Contraction: A mapping $\mathbf{T} : \mathcal{X} \mapsto \mathcal{X}$ where $\mathcal{X}$ is a subset of a normed space N is called a *contraction mapping*, or simply a *contraction*, if there is a positive number $a < 1$ such that

$$\|\mathbf{T}\mathbf{k}_1 - \mathbf{T}\mathbf{k}_2\| \leq a\|\mathbf{k}_1 - \mathbf{k}_2\|, \quad \forall \mathbf{k}_1\mathbf{k}_2 \in \mathcal{X}. \qquad (20)$$

The definition is central to

Theorem A.1. Contraction Mapping Theorem*: If* $\mathbf{T} : \mathcal{X} \mapsto \mathcal{X}$ *is a contraction mapping a closed subset* $\mathcal{X}$ *of a Banach space, then there is exactly one* $\mathbf{x} \in \mathcal{X}$ *such that* $\mathbf{T}\mathbf{x} = \mathbf{x}$*. For any* $\mathbf{x}_0 \in \mathcal{X}$*, the sequence* $(\mathbf{x}_n)$ *defined by* $\mathbf{x}_{n+1} = \mathbf{T}\mathbf{x}_n$ *converges to* $\mathbf{x}$*.*

The challenge is thus to determine a contraction $\mathbf{T}$ with a suitable fixed point, i.e. a fixed point solution which

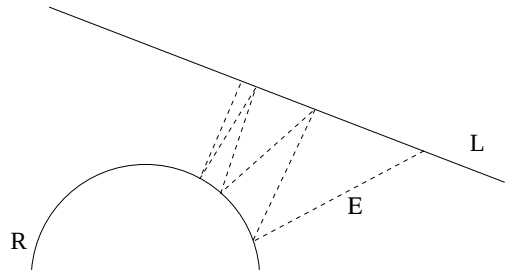


Figure 15: **Geometrical interpretation**: R represents the manifold of calibrated cameras assuming no error on the intrinsic parameters. L represents the linear subspace covered by H. E is the error on the intrinsic parameters with $\|\mathrm{E}\|$ being the Euclidian distance between corresponding representatives R and KR on the respective manifolds. The dashed line represents the iterations performed (3 iterations shown).

minimises the sum of squared errors between the $3m$ estimated intrinsic parameters and the wanted parameters from (8).

A geometrical interpretation is given in Figure 15, where the manifolds to which $\mathcal{R}$ and $\mathcal{KR}$ belong are denoted R and L respectively. L is simply the linear subspace covered by $\mathbf{H}$ whereas R is clearly non-linear and would intuitively appear as related to a sphere. In the figure, three iterations are shown, beginning at some starting point on L. It is of particular interest to note that the error E is simply the Euclidian distance between the two corresponding representatives R and KR on R and L. Minimising the error thus amounts to find the point on the manifolds where this distance is minimised, proceeding iteratively as shown on the figure.

A.1.1 Constructing a Contraction T

Essentially, we construct $\mathbf{T}$ as a mapping that takes all the involved forms (7) closer to their forms (8). The characteristic that will be central to constructing $\mathbf{T}$ is that $\mathbf{r}_1$ and $\mathbf{r}_2$ are orthonormal in (8). We will thus only consider the upper left 2×3 blocks of the affine camera matrices. Also, it will be practical to visualise $\mathbf{r}_1$ and $\mathbf{r}_2$ as orthogonal points on $\mathbb{S}^2$. We will then consider a transformation $\mathbf{T}$ that gradually enforces pairwise orthonormality on all these $2m$ 3D points corresponding to the m affine cameras and conjecture that it is a contraction.

We consider the mapping

$$\mathbf{T} : \mathbb{S}^{3m-1}(\sqrt{2m}) \quad \mapsto \quad \mathbb{S}^{3m-1}(\sqrt{2m}), \qquad (21)$$

where $\mathbb{S}^{3m-1}(\sqrt{2m})$ denotes the $(3m-1)$-dimensional hypersphere in $\mathtt{R}^{\mathtt{3m}}$ with radius $\sqrt{2m}$ (the choice of $\mathcal{X} = \mathbb{S}^{3m-1}(\sqrt{2m})$ will become clear shortly). For completeness we point out that $\mathbb{S}^{3m-1}(\sqrt{2m})$ is a closed subset of a Banach space: the embedding vector space $\mathbb{R}^{3m}$ is normed and complete and the sphere is indeed closed.

Let $\mathbf{k} \in \mathbb{S}^{3m-1}(\sqrt{2m})$ be the vector containing the three affine intrinsic calibration parameters $\hat{\gamma_x}$, $\hat{\gamma_y}$ and s, normalised to $\|\mathbf{k}\|_2 = \sqrt{2m}$. $\mathbf{k}$ has the form

$$\mathbf{k} = \begin{bmatrix} \hat{\gamma}_{x1} \\ \hat{\gamma}_{y1} \\ s_1 \\ \vdots \\ \hat{\gamma}_{xm} \\ \hat{\gamma}_{ym} \\ s_m \end{bmatrix} \frac{1}{s_{\sqrt{2m}}}. \qquad (22)$$

where $s_{\sqrt{2m}}$ is a scale factor ensuring $\mathbf{k} \in \mathbb{S}^{3m-1}(\sqrt{2m})$

We introduce $\mathcal{P}$ as the $2m \times 3$ stack of all the 2×3 matrices $\bar{\mathbf{P}}$ from (7), i.e.

$$\mathcal{P} = \begin{bmatrix} \bar{\mathbf{P}}_1 \\ \vdots \\ \bar{\mathbf{P}}_m \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{K}}_1\bar{\mathbf{R}}_1 \\ \vdots \\ \bar{\mathbf{K}}_m\bar{\mathbf{R}}_m \end{bmatrix},$$

where the right-hand side is a *RQ-factorisation* of each of the $\bar{\mathbf{P}}$'s, i.e. $\bar{\mathbf{K}}_i$ is 2×2 upper triangular containing 3 entries form $\mathbf{k}$ and $\mathbf{R}_i$ has two orthonormal rows. Furthermore, we define the block-diagonal $2m \times 2m$ matrix $\mathcal{K}$ and the $2m \times 3$ matrix $\mathcal{R}$ such that

$$\mathcal{K} = \begin{bmatrix} \bar{\mathbf{K}}_1 & & \\ & \ddots & \\ & & \bar{\mathbf{K}}_m \end{bmatrix}, \quad \mathcal{R} = \begin{bmatrix} \bar{\mathbf{R}}_1 \\ \vdots \\ \bar{\mathbf{R}}_m \end{bmatrix},$$

i.e.

$$\mathcal{P} = \mathcal{KR}.$$

Note that the pairs of rows of $\mathcal{R}$ are orthonormal and that the structure of each of the point pairs in $\mathcal{P}$ is encoded in $\mathcal{K}$. Also note that the representation of $\mathcal{R}$ on

$\mathbb{S}^{3m-1}(\sqrt{2m})$ would be

$$\rho = \left.\begin{bmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 1 \\ 1 \\ 0 \end{bmatrix}\right\} \; m \text{ repetitions of } [1\ 1\ 0]^\top, \qquad (23)$$

i.e. ρ is a $3m \times 1$ vector with norm $\sqrt{2m}$ containing the ideal calibration parameter values. ρ would thus be a perfect fixed point, however, it is in the general case not attainable since that would require complete absence of noise.

Let $\mathbf{H}$ denote the affine transformation that minimises the distance between $\mathcal{P}$ and $\mathcal{R}$ i.e.

$$\min_{\mathbf{H}} \|\mathcal{P}\mathbf{H} - \mathcal{R}\|_F,$$

where $\|\mathbf{A}\|_F$ denotes the Frobenius-norm, i.e. the square root of the sum of the squares of all the elements of the matrix $\mathbf{A}$. Note that $\mathbf{H}$ may be considered as the affine transformation that optimally aligns the point clouds described by the rows of the two $2m \times 3$ matrices $\mathcal{P}$ and $\mathcal{R}$ and let

$$\hat{\mathcal{P}} = \mathcal{P}\mathbf{H} = \hat{\mathcal{K}}\hat{\mathcal{R}} \qquad (24)$$

denote the optimally aligned point cloud. By extracting the new estimate of the intrinsic parameters from $\hat{\mathcal{K}}$, denoted $\hat{\mathbf{k}}$, the output of $\mathbf{T}$ is obtained.

A.1.2 Some Constraints on H

For the sake of the present demonstration, i.e. in order to obtain a unique fixed point, we need to eliminate the 3-parameter ambiguity stemming from the undetermined global rotation. This may be done by fixing the three parameters of the first camera's rotation, for instance

$$\bar{\mathbf{R}}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \qquad (25)$$

We thus consider the 5-dimensional subset $\mathcal{V}$ of the $6m - 3 - 1$ dimensional vector space of all the possible $\mathcal{P}$ with fixed first camera ($6m$ for the coefficients of all the 2×3 blocks, -3 for fixing the first camera and -1 for fixing overall scale). It is simple to see by inspection that $\mathbf{H}$ should have the form

$$\mathbf{H} = \begin{bmatrix} * & * & 0 \\ 0 & * & 0 \\ * & * & * \end{bmatrix} \frac{1}{s'_{\sqrt{2m}}} \qquad (26)$$

in order to preserve (25), $s'_{\sqrt{2m}}$ ensuring the right overall scale. $*$ indicates a possibly non-zero entry.

We now conjecture that aligning two point clouds $\mathcal{K}_1\mathcal{R}_1 \in \mathcal{V}$ and $\mathcal{K}_2\mathcal{R}_2 \in \mathcal{V}$ to the point clouds $\mathcal{R}_1$ and $\mathcal{R}_2$ which have the local structure (orthogonal point pairs) brings each of the local structures closer to each other from an overall least squares point of view, i.e.

$$\|\mathbf{T}\mathbf{k}_1 - \mathbf{T}\mathbf{k}_2\|_2 \leq \alpha\|\mathbf{k}_1 - \mathbf{k}_2\|_2, \qquad (27)$$

with $\alpha < 1$. As formulated in (27), the conjecture seems to be somewhat false and overly general. However, the precise delimitation is not very clear, and it experimentally appears to be sufficiently true within the domain required by our application. In the experiment, a set of $m = 10$ cameras where generated randomly with intrinsic parameters normalised to $\hat{\gamma}_x = \hat{\gamma}_y = 1$ and $s = 0$. Additive Gaussian noise with standard deviation σ was applied to the parameters, and two instances $\mathcal{K}_1\bar{\mathbf{R}}_1$ and $\mathcal{K}_1\bar{\mathbf{R}}_1$ and their corresponding $\mathbf{k}_1$ and $\mathbf{k}_2$ were created by applying two random transformations $\mathbf{H}_1$ and $\mathbf{H}_2$ of the form (26). They were subsequently transformed by $\mathbf{T}$ and it was verified if they fulfilled (27). The results are shown in the lower curve in Figure 16 for a varying noise level. It was experimentally found (Section 6) that a typical autocalibration application will have a noise level $\sigma \approx 0.25$ corresponding to a probability of $\approx .97$ to fulfill the conjecture. Note also that lying within the conjecture as we will see guarantees an optimal estimate, however the optimal estimate might very well be reached without the conjecture being satisfied. This is illustrated by the upper curve, which is the percentage of tries where the algorithm converged to the desired minimum. The curve lies steadily at a 100% success rate.

A.1.3 Validity of the Fixed Point

Equation (27) implies that $\mathbf{T}$ is a contraction. It follows from the Contraction Mapping Theorem that the equation $\mathbf{k} = \mathbf{T}\mathbf{k}$ has a unique solution $\mathbf{k}_0$. It now remains to be

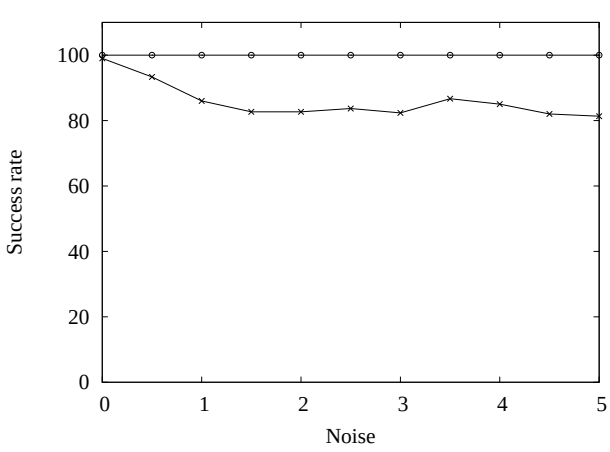


Figure 16: **Validity of the conjecture in equation (27) and convergence to a fixed point.** Graph showing the validity of the conjecture (lower curve) and the percentage of succesful convergence of the algorithm to a fixed point (upper curve), when applied to the autocalibration problem with varying noise level. The x-axis indicates the standard deviation of the noise applied to the normalised intrinsic parameters $(\hat{\gamma}_x, \hat{\gamma}_y, s) = (1, 1, 0)$. The y-axis indicates the percentage of times the conjecture is true (lower curve) and convergence to the desired fixed point (upper curve). The typical autocalibration application has $\sigma \approx 0.25$. The upper curve indicates a 100% success rate (convergence to the desired minimum) of the algorithm.

proven that the fixed point $\mathbf{k}_0$ is a Least Squares Estimate of the affine intrinsic parameters (up to an undefined scale factor).

Since $\mathbf{k}_0$ is a fixed point $\mathbf{k}_0 = \mathbf{T}\mathbf{k}_0$ and accordingly $\mathbf{H} = \mathbf{I}_{3\times 3}$ (according to (24)) and the point clouds $\mathcal{K}_0\mathcal{R}_0$ and $\mathcal{R}_0$ are thus optimally aligned, i.e.

$$\|\mathcal{K}_0\mathcal{R}_0 - \mathcal{R}_0\|_F = \min_{\mathcal{K}\mathcal{R}\in\mathcal{V}} \|\mathcal{K}\mathcal{R} - \mathcal{R}\|_F \ . \qquad (28)$$

Also, considering each pair of points separately and decomposing each $\bar{\mathbf{R}}_i$ into

$$\bar{\mathbf{R}}_i = \begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{bmatrix}$$

and each $\bar{\mathbf{K}}_i - \mathbf{I}$ into

$$\bar{\mathbf{K}}_i - \mathbf{I} = \begin{bmatrix} k_1 & k_3 \\ 0 & k_2 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \epsilon_1 & \epsilon_3 \\ 0 & \epsilon_2 \end{bmatrix},$$

we see that

$$\begin{aligned} \|\bar{\mathbf{K}}_i\bar{\mathbf{R}}_i - \bar{\mathbf{R}}_i\|_F^2 &= \|(\bar{\mathbf{K}}_i - \mathbf{I})\bar{\mathbf{R}}_i\|_F^2 \\ &= \epsilon_1^2(u_1^2 + u_2^2 + u_3^2) \\ &+ \epsilon_2^2(v_1^2 + v_2^2 + v_3^2) \\ &+ \epsilon_3^2(v_1^2 + v_2^2 + v_3^2) \\ &+ 2\epsilon_1\epsilon_3(u_1v_1 + u_2v_2 + u_3v_3) \\ &= \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2, \end{aligned} \qquad (29)$$

which is valid for every of the m point pairs. Thus the minimisation in (28) which is the one that is performed, is equivalent to the minimisng (29) for all the $i = 1..m$ cameras, which affirms the estimate as a Least Squares Estimate.

Finally, if the estimated $\mathbf{H}$ at each iteration is denoted $\mathbf{H}_j$, $j = 1.., m_i$, j denoting the iteration number and m_i the number of iterations, the upgrading (calibrating) transformation $\mathbf{H}_c$ is obtained as

$$\mathbf{H}_c = \prod_{j=1}^{m_i} \mathbf{H}_j$$

and is applied to each camera $\mathbf{P}'_i$ according to

$$\mathbf{P}_{Ai} = \mathbf{P}'_i\mathbf{H}_c \ . \qquad (30)$$

Acknowledgements

The authors wish to thank Bill Triggs for valuable inspiration concerning the $\mathbf{F}_A$-closure and its application it in a useful way, and Eric Hayman for valuable comments of the proposed ideas.

References

[1] H. Aanaes, R. Fisker, K. Astrom, and J.M. Carstensen. Robust factorization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(9):1215–1225, 2002.

[2] P. Beardsley, A. Zisserman, and D. W. Murray. Sequential updating of projective and affine structure from motion. *Int. Journal of Computer Vision*, 23(3):235–259, 1997.

[3] Å. Björck. *Numerical methods for least squares problems*. Society for Industrial and Applied Mathematics, 1996.

[4] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1098–1104, 1996.

[5] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Computer Vision - ECCV92*, volume 588 of *Lecture Notes in Computer Science*, pages 563–578. Springer Verlag, 1992.

[6] O. Faugeras. *Three-dimensional vomputer vision*. MIT Press, Cambridge, Mass, 1993.

[7] D. H. Griffel. *Applied functional analysis*. Dover Publications, 1985.

[8] N Guilbert. A structure from motion system in octave. *http://www.maths.lth.se/∼nicolas*, 2006.

[9] N. Guilbert and A. Bartoli. Batch recovery of multiple views with missing data using direct sparse solvers. In *Proc. British Machine Vision Conference*, volume 1, pages 63–72, 2003.

[10] N. Guilbert, F. Kahl, M. Oskarsson, K. Åström, M. Johansson, and A. Heyden. Constraint enforcement in structure from motion applied to closing an open sequence. In *Proc. Asian Conf. on Computer Vision, Jeju Island, Korea*, 2004.

[11] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In Joseph L. Mundy, Andrew Zisserman, and David Forsyth, editors, *Applications of Invariance in Computer Vision*, volume 825 of *Lecture notes in Computer Science*, pages 237–256. Springer-Verlag, 1994.

[12] R. I. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[13] A. Heyden and K Åström. Flexible calibration: Minimal cases for auto-calibration. In *Proc. 7th Int. Conf. on Computer Vision, Kerkyra, Greece*, 1999.

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

[15] F. Kahl and A. Heyden. Affine structure and motion from points, lines and conics. *Int. Journal of Computer Vision*, 1999.

[16] T. Kanade and D. Morris. Factorization methods for structure from motion. *Phil. Trans. R. Soc. Lond.*, A(356):1153–1173, 1998.

[17] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[18] D. Martinec and T. Pajdla. Structure from many perspective images with occclusions. In *Proc. 7th European Conf. on Computer Vision, Copenhagen, Denmark*, 2002.

[19] D. Nistér. *Automatic dense reconstruction from uncalibrated video sequences*. PhD thesis, Dept. of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, 2001.

[20] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(3):206–218, march 1997.

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

[22] L. Quan. Self-calibration of an affine camera from multiple views. *Int. Journal of Computer Vision*, 19(1):93–105, 1996.

[23] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, June 1994.

[24] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. Journal of Computer Vision*, 9(2):137–154, 1992.

[25] B. Triggs. Linear projective reconstruction from matching tensors. *Image and Vision Computing*, 15(8):617–625, 1997.

[26] G. Xu and Z Zhang. *Epipolar geometry in stereo, motion and object recognition. A unified approach*. Springer Verlag, 1996.

[27] Zhang and Xu. A unified theory of uncalibrated stereo for both perspective and affine cameras. *Journal of Mathematical Imaging and Vision*, 9(3):213–229, 1998.