A Generic Calibration Concept : Theory and Algorithms

Peter Sturm, Srikumar Ramalingam

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Generic Calibration Concept – Theory and Algorithms

Peter Sturm — Srikumar Ramalingam

N° 5058

December 2003

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--5058--FR+ENG

INRIA
RHÔNE-ALPES

A Generic Calibration Concept – Theory and Algorithms

Peter Sturm* , Srikumar Ramalingam †

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Movi

Rapport de recherche n° 5058 — December 2003 — 40 pages

Abstract: We present a theory and algorithms for a generic calibration concept that is based on the following recently introduced general imaging model. An image is considered as a collection of pixels, and each pixel measures the light traveling along a (half-) ray in 3-space associated with that pixel. Calibration is the determination, in some common coordinate system, of the coordinates of all pixels' rays. This model encompasses most projection models used in computer vision or photogrammetry, including perspective and affine models, optical distortion models, stereo systems, or catadioptric systems, central (single viewpoint) as well as non-central ones. We propose a concept for calibrating this general imaging model, based on several views of objects with known structure, but which are acquired from unknown viewpoints. It allows to calibrate cameras of any of the types contained in the general imaging model using one and the same algorithm. We first develop the theory and an algorithm for the most general case: a non-central camera that observes 3D calibration objects. This is then specialized to the case of central cameras and to the use of planar calibration objects. For calibrated images, we have also demonstrated the applications in perspective view synthesis, epipolar curve analysis and structure from motion algorithms. The validity of the concept is shown by experiments with synthetic and real data.

Key-words: Calibration, camera model, distortion, stereo, omnidirectional, catadioptric

* Peter.Sturm@inrialpes.fr
† srikumar@cse.ucsc.edu – S. Ramalingam is with the University of California at Santa Cruz.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 Montbonnot-St-Martin (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Un concept générique pour le calibrage – Théorie et algorithmes

Résumé : Nous présentons une théorie et des algorithmes pour un concept générique pour le calibrage de caméras. Ce concept est basé sur un modèle de création d'images général : une image est considérée comme une collection de pixels et chaque pixel « mesure la lumière » qui « voyage » le long d'un rayon (demi-droite) en 3D qui est associée à ce pixel. Le calibrage consiste alors en la détermination, dans un repère quelconque, des coordonnées des rayons de tout les pixels. Ce modèle comprend quasiment tous les modèles de caméra couramment utilisés en vision par ordinateur ou photogrammétrie, y inclus les projections perspectives et affines, des modèles de distorsions optiques, les systèmes stéréo, ou bien des capteurs catadioptriques, aussi bien des capteurs « centraux » (à point de vue effectif unique) que non-centraux. Nous proposons un concept pour calibrer ce modéle de création d'images général, qui utilise plusieurs images d'un objet de calibrage avec structure connue, mais prises de points de vue inconnues. Ceci permet de calibrer toute caméra contenue dans notre modèle général, avec le même algorithme ! Nous dévelopons d'abord la théorie et un algorithme pour le cas le plus général : calibrage d'une caméra non-centrale avec un object de calibrage volumique. Ceci est ensuite spécialisé au cas des caméras centrales et à l'utilisation d'objets de calibrage plans. Nous discutons également des applications du calibrage de ces caméras générales, concernant la synthèse de vues perspectives, la géométrie épipolaire et l'estimation du mouvement et de la structure de la scène. Des expériences préliminaires sur données synthétiques et réelles montrent la validité de notre concept.

Mots-clés : Calibrage, modèle de caméra, distorsion, stéréo, omnidirectionnel, catadioptrique

1 Introduction

We consider the camera calibration problem, i.e. the estimation of a camera's intrinsic parameters. A camera's intrinsic parameters (plus the associated projection model) give usually exactly the following information: for any point in the image, they allow to compute a ray in 3D along which light travels that falls onto that point (here, we neglect point spread).

Consider for example the pinhole model, i.e. a camera with calibration matrix K and projection matrix

$$\mathsf{P} = \mathsf{K}\mathsf{R}\begin{pmatrix}\mathtt{I} & -\mathbf{t}\end{pmatrix} \quad .$$

Decompose into a canonic projection matrix (depending only on intrinsic parameters) and a pose transformation:

$$\mathsf{P} = \begin{pmatrix}\mathsf{K} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1\end{pmatrix}$$

For a point $\mathbf{q}$ in the image (in homogeneous coordinates), the canonic projection matrix allows to determine the ray along which the corresponding 3D point $\mathbf{Q}$ must lie. Relative to the camera, the ray can for example be defined by the optical center (the origin in the local coordinate frame) and the point at infinity along the ray, computed as:

$$\mathbf{Q}_\infty = \begin{pmatrix}\mathsf{K}^{-1}\mathbf{q} \\ 0\end{pmatrix}$$

To get the ray in the absolute coordinate frame, these two points should be transformed by the inverse of the above pose transformation.

Other common camera models perform basically the same task: given a point in the image, compute a ray in 3D. Examples are the pinhole model augmented with radial or other distortion parameters, models for omni-directional cameras, for non-central mosaics, etc. These models allow of course also the "inverse" operation, i.e. projecting a 3D point onto an image point.

Most existing camera models are parametric (i.e. defined by a few intrinsic parameters) and address imaging systems with a single effective viewpoint (all rays pass through one point). In addition, existing calibration procedures are taylor-made for specific camera models.

The aim of this work is to relax the last two constraints: we want to propose and develop a calibration method that should work for any type of camera model, and especially also for cameras without a single viewpoint. To do so, we first renounce on parametric models, and adopt the following very general model: a camera acquires images consisting of pixels; each pixel captures light that travels along a ray in 3D. The camera is fully described by:

- the coordinates of these rays (given in some local coordinate frame).
- the mapping between rays and pixels; this is basically a simple indexing.

The non-parametric nature of this model adds one difficulty: how to compute 3D rays for an image point with non-integer image coordinates. To do so, the only possibility is to add continuity assumptions, e.g. that neighboring pixels have neighboring 3D rays. Under this or more restrictive assumptions, 3D rays for arbitrary image points can be computed by interpolation. Similarly, the projection of 3D points onto images, is not straightforward, but can for example be solved analogously, by interpolation. For the time being, we do not address these issues in more detail.

The above general camera model allows to describe virtually any camera that captures light rays traveling along straight lines [1]. Examples:

- a camera with any type of optical distortion, such as radial or tangential.
- a camera looking at a reflective surface, e.g. as often used in surveillance, a camera looking at a spherical or otherwise curved mirror [9]. Such systems, as opposed to central catadioptric systems [2] composed of cameras and parabolic mirrors, do not in general have a single viewpoint.
- multi-camera stereo systems: put together the pixels of all image planes; they "catch" light rays that definitely do not travel along lines that all pass through a single point. Nevertheless, in the above general camera model, a stereo system (with rigidly linked cameras) is considered as a **single** camera.
- other acquisition systems, see e.g. [3, 13, 18], or eyes of some insects.

Relation to previous work. See [8, 16] for reviews and references on existing calibration methods and e.g. [5] for an example related to catadioptric devices.

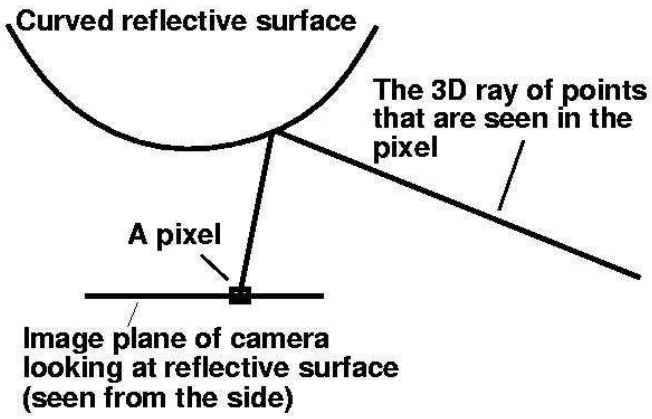


Figure 1: Sketch of a cut through a catadioptric system. Note that the 3D rays do not pass through their associated pixels here.

The above imaging model has already been used, in more or less explicit form, in various works [7, 11, 12, 13, 14, 15, 18, 20, 21, 22], and is best described in [7]. There are conceptual links to other works: acquiring an image with a camera of our general model may be seen as sampling the plenoptic function [1], and a light field [10] or lumigraph [6] may be interpreted as a single image, acquired by a camera of an appropriate design.

To our knowledge, the only previously proposed calibration approach for the general imaging model, is due to Grossberg and Nayar [7]. Their approach requires the acquisition of two or more images of a calibration object with known structure, and knowledge of the camera or object motion between the acquisitions. In this work, we develop a completely general approach,

[1] However, it would not work for example with a camera looking from the air, into water: still, to each pixel is associated a refracted ray in the water. However, when the camera moves, the refraction effect causes the set of rays to not move rigidly, hence the calibration would be different for each camera position.

that requires taking three or more images of calibration objects, from arbitrary and **unknown viewing positions**. We develop variants of the general concept, according to using a central vs. the general non-central camera model and to using a planar vs. a 3D calibration object.

Other related work deals mostly with epipolar geometry estimation and modeling [12, 15, 21] and motion estimation for already calibrated cameras [11, 14].

Organization. In section 2, we explain the camera model used in this work and give some notations. For ease of explanation and understanding, the calibration concept is first introduced for 2D cameras, in section 3. The general concept for 3D cameras is described in section 4 and variants of the concept (central camera and planar calibration objects) are dealt with in section 5. Some experimental results are shown in section 6, followed by discussions and conclusions. In appendices A and B, we investigate under which conditions stereo systems consisting of several central cameras, can be calibrated using the generic algorithm.

2 Camera Model and Notations

We give here the definition of the (purely geometrical) camera model used in this work. It is essentially the same as the model described in [7] where in addition other issues such as point spread and radiometry are treated as well.

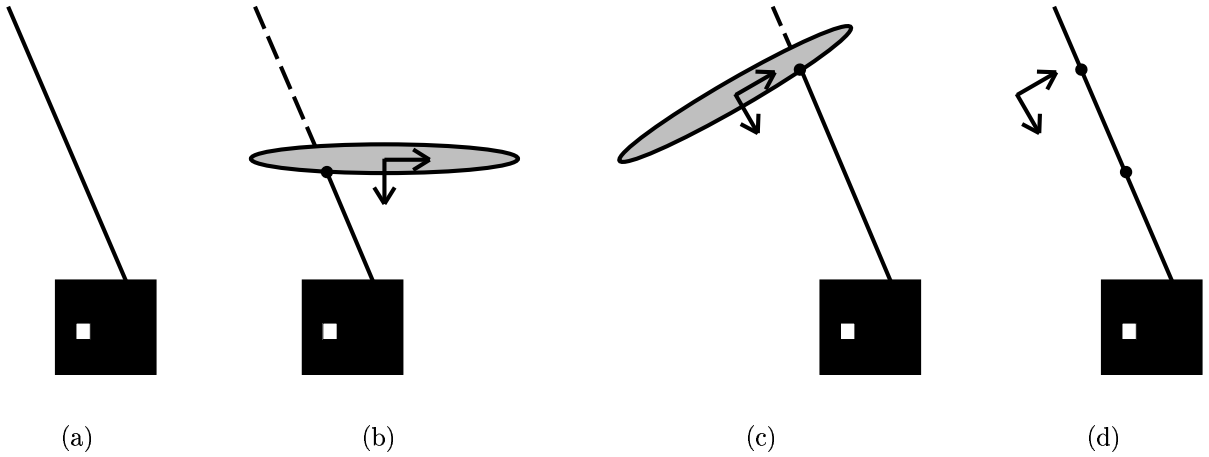
We assume that a camera delivers images that consist of a set of pixels, where each pixel captures/measures the light traveling along some half-ray. In our calibration method, we do not model half-rays explicitly, but rather use their infinite extensions – **camera rays**. Camera rays corresponding to different pixels need not intersect – in this general case, we speak of **non-central cameras**, whereas if all camera rays intersect in a single point, we have a **central camera** with an **optical center**.

Furthermore, the physical location of the actual photosensitive elements that correspond to the pixels, does in general not matter at all. On the one hand, this means that the camera ray corresponding to some pixel, needs not pass through that pixel, cf. figure 1. On the other hand, neighborship relations between pixels are in general not necessary to be taken into account: the set of a camera's photosensitive elements may lie on a single surface patch (image plane), but may also lie on a 3D curve, on several surface patches or even be placed at completely isolated positions. In practice however, we do use some continuity assumption, useful in the stage of 3D-2D matching, as explained in section 6: we suppose that pixels are indexed by two integers (the pixel's *coordinates*) like in traditional cameras and that pixels with neighboring coordinates have associated camera rays that are "close" to one another.

3 The Calibration Concept for 2D Cameras

We consider here a camera and scene living in a 2D plane, i.e. camera rays are lines in that plane.

Input. We acquire two images of an object undergoing some motion. Consider here a single pixel and its camera ray, as illustrated in figure 2. Figures 2 (b) and (c) show the two points on the object that are seen by that pixel in the two images. We suppose to be able to determine the coordinates of these two points, first in some local coordinate frame attached to the object ("matching").



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

Figure 2: (a) The camera as black box, with one pixel and its camera ray. (b) The pixel sees a point on a calibration object, whose coordinates are identified in a frame associated with the object. (c) Same as (b), for another position. (d) Due to known motion, the two points on the calibration object can be placed in the same coordinate frame. The camera ray is the determined by joining them.

The case of known motion. If the object's motion between the image acquisitions is known, then the two object points can be mapped to a single coordinate frame, e.g. the object's coordinate frame for the second image, as shown in figure 2 (d). Computing our pixel's camera ray is then simply done by joining the two points. This summarizes the calibration approach proposed by Grossberg and Nayar [7], applied here for the 2D case. Camera rays are thus initially expressed in a coordinate frame attached to one of the calibration object's positions. This does not matter (all that counts are the relative positions of the camera rays), but for convenience, one would typically try to choose a better frame. For a central camera for example, one would choose the optical center as origin or for a non-central camera, the point that minimizes the sum of distances to the set of camera rays (if it exists).

The case of unknown motion. The above approach is no longer applicable, and we need to estimate, implicitly or explicitly, the unknown motion. We now show how to do this, given three images. Let us note the three points on the calibration objects, that are seen in the same pixel, by $\mathbf{Q}, \mathbf{Q}'$ and $\mathbf{Q}''$. These are 3-vectors of homogeneous coordinates, expressed in the respective local coordinate frame. Without loss of generality, we choose the coordinate frame associated with the object's first position, as common frame. The unknown (relative) motions, that allow to map the second and third frames onto the first one, are given by 2×2 rotation matrices R' and R'' and translation vectors $\mathbf{t}'$ and $\mathbf{t}''$. Note that for the rotation matrices we have $R'_{11} = R'_{22}$ and $R'_{12} = -R'_{21}$ (and similarly for R''). The calibration points, after mapping to the common frame, are given as:

$$\mathbf{Q} \qquad \begin{pmatrix} \mathsf{R}' & \mathbf{t}' \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \mathbf{Q}' \qquad \begin{pmatrix} \mathsf{R}'' & \mathbf{t}'' \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \mathbf{Q}''$$

They must all lie on the pixel's camera ray, hence they must be collinear. Algebraically, this is expressed by the fact that the determinant of the 3×3 matrix composed of the above three

coordinate vectors, vanishes:

$$\begin{vmatrix} Q_1 & R'_{11}Q'_1 + R'_{12}Q'_2 + t'_1Q'_3 & R''_{11}Q''_1 + R''_{12}Q''_2 + t''_1Q''_3 \\ Q_2 & R'_{21}Q'_1 + R'_{22}Q'_2 + t'_2Q'_3 & R''_{21}Q''_1 + R''_{22}Q''_2 + t''_2Q''_3 \\ Q_3 & Q'_3 & Q''_3 \end{vmatrix} = 0 \quad (1)$$

This equation is trilinear in the calibration point coordinates. The equation's coefficients may be interpreted as coefficients of a trilinear matching tensor; they depend on the unknown motions' coefficients, and are given in table 1. In the following, we sometimes call this the "calibration tensor". It is somewhat related to the *homography tensor* derived in [17].

i	C_i	V_i
1	$Q_1Q'_1Q''_3 + Q_2Q'_2Q''_3$	R'_{21}
2	$Q_1Q'_2Q''_3 - Q_2Q'_1Q''_3$	R'_{22}
3	$Q_1Q'_3Q''_1 + Q_2Q'_3Q''_2$	$-R''_{21}$
4	$Q_1Q'_3Q''_2 - Q_2Q'_3Q''_1$	$-R''_{22}$
5	$Q_3Q'_1Q''_1 + Q_3Q'_2Q''_2$	$R'_{11}R''_{21} - R''_{11}R'_{21}$
6	$Q_3Q'_1Q''_2 - Q_3Q'_2Q''_1$	$R'_{11}R''_{22} - R''_{12}R'_{21}$
7	$Q_1Q'_3Q''_3$	$t'_2 - t''_2$
8	$Q_2Q'_3Q''_3$	$-t'_1 + t''_1$
9	$Q_3Q'_1Q''_3$	$R'_{11}t''_2 - R'_{21}t''_1$
10	$Q_3Q'_2Q''_3$	$R'_{12}t''_2 - R'_{22}t''_1$
11	$Q_3Q'_3Q''_1$	$R''_{21}t'_1 - R''_{11}t'_2$
12	$Q_3Q'_3Q''_2$	$R''_{22}t'_1 - R''_{12}t'_2$
13	$Q_3Q'_3Q''_3$	$t'_1t''_2 - t''_1t'_2$

Table 1: Coupled variables in the trifocal calibration tensor for a general 2D camera. Coefficients not shown here are always zero.

Among the $3 \cdot 3 \cdot 3 = 27$ coefficients of the calibration tensor, 8 are always zero and among the remaining 19 ones, there are 6 pairs of identical ones. The columns of table 1 are interpreted as follows: the C_i are trilinear products of point coordinates and the V_i are the associated coefficients of the tensor. The following equation is thus equivalent to (1):

$$\sum_{i=1}^{13} C_i V_i = 0 \quad (2)$$

Given triplets of points $\mathbf{Q}, \mathbf{Q}'$ and $\mathbf{Q}''$ for at least 12 pixels, we may compute the trilinear tensor up to an unknown scale λ by solving a system of linear equations of type (2). Note that we have verified using simulated data, that we indeed can obtain a unique solution (up to scale) for the tensor. The main questions now are, if the motion coefficients can be extracted from the tensor. This is indeed possible, as shown below. Once the motions are determined, the above approach can be readily applied to compute the camera rays.

We now describe an algorithm for extracting the motion parameters. Let the estimated tensor coefficients be V'_i; they are equal to the coefficients of table 1 up to an unknown scale: $V'_i = \lambda V_i, i = 1 \ldots 13$. The algorithm works as follows:

1. Estimate λ: $\lambda = \sqrt{V'^2_1 + V'^2_2}$ (orthonormality of 2×2 rotation matrix R').

2. Compute $V_i = \frac{V'_i}{\lambda}, i = 1 \ldots 13$.
3. Compute R': $R'_{11} = R'_{22} = V_2$ and $R'_{21} = -R'_{12} = V_1$
4. Compute R'': $R''_{11} = R''_{22} = -V_4$ and $R''_{21} = -R''_{12} = -V_3$
5. Compute $\mathbf{t}''$: $t''_1 = -V_9 R'_{21} - V_{10} R'_{22}$ and $t''_2 = \frac{V_9 + R'_{21} t'_1}{R'_{11}}$
6. Compute $\mathbf{t}'$: $t'_1 = t''_1 - V_8$ and $t'_2 = V_7 + t''_2$

The correctness of the method was verified by simulated random experiments.

The special case of central cameras. It is worthwhile to specialize the calibration concept to the case of central cameras (but which are otherwise general, i.e. not perspective). A central camera can actually already be calibrated from two views, as explained in the following. Let $\mathbf{C}$ be the homogeneous coordinates of the optical center (given in the frame associated with the object's first position). We have the following collinearity constraint:

$$\begin{vmatrix} C_1 & Q_1 & R'_{11}Q'_1 + R'_{12}Q'_2 + t'_1 \\ C_2 & Q_2 & R'_{21}Q'_1 + R'_{22}Q'_2 + t'_2 \\ C_3 & Q_3 & Q'_3 \end{vmatrix} = 0$$

which can be written as

$$\mathbf{Q}'^{\mathsf{T}} \begin{pmatrix} R'_{21}C_3 & -R'_{22}C_3 & R'_{22}C_2 - R'_{21}C_1 \\ R'_{22}C_3 & R'_{21}C_3 & -R'_{21}C_2 - R'_{22}C_1 \\ C_3 t'_2 - C_2 & C_1 - C_3 t'_1 & C_2 t'_1 - C_1 t'_2 \end{pmatrix} \mathbf{Q} = 0$$

The bifocal calibration tensor in this equation is a 3×3 matrix, that is somewhat similar to a fundamental or essential matrix. It is of rank 2 and its right null vector is the optical center $\mathbf{C}$, which can thus be easily extracted.

The special case of a linear calibration object. It is equally worthwhile to specialize our concept to the case of a linear calibration object. We now consider again the general, non-central camera model. With loss of generality, we suppose that the calibration points lie on a line with $Y = 0$ in the local coordinate frame, i.e. $Q_2 = Q'_2 = Q''_2 = 0$. The collinearity equation (1) gives then rise to a $2 \times 2 \times 2$ trifocal calibration tensor. Among its 8 coefficients, only 1 is always zero, and among the others, none are identical to one another, cf. table 2.

We observe that the rotation coefficients R'_{22} and R''_{22} do not appear individually, contrary to the tensor for the general case. Hence, the scale factor λ can no longer be determined as easily as in the above algorithm. The motion parameters can nevertheless be extracted, but the algorithm is more complicated, and not shown here for lack of space.

Another special case not shown here is that of calibrating a central camera using a linear calibration object.

i	C_i	V_i
1	$Q_1 Q'_1 Q''_3$	R'_{21}
2	$Q_1 Q'_3 Q''_1$	$-R''_{21}$
3	$Q_1 Q'_3 Q''_3$	$t'_2 - t''_2$
4	$Q_3 Q'_1 Q''_1$	$R'_{22}R''_{21} - R'_{21}R''_{22}$
5	$Q_3 Q'_1 Q''_3$	$R'_{22}t''_2 - R'_{21}t''_1$
6	$Q_3 Q'_3 Q''_1$	$R''_{21}t'_1 - R''_{22}t'_2$
7	$Q_3 Q'_3 Q''_3$	$t'_1 t''_2 - t''_1 t'_2$

Table 2: Coefficients of the trifocal calibration tensor for a general 2D camera and a linear calibration object.

4 Generic Calibration Concept for 3D Cameras

This and the next section describe our main contributions. We extend the concept described in section 3 to the case of cameras living in 3-space. We first deal with the most general case: non-central cameras and 3D calibration objects.

In case of **known motion**, two views are sufficient to calibrate, and the procedure is equivalent to that outlined in section 3, cf. [7]. In the following, we consider the practical case of **unknown motion**. Input are now, for each pixel, three 3D points $\mathbf{Q}, \mathbf{Q}'$ and $\mathbf{Q}''$, given by 4-vectors of homogeneous coordinates, relative to the calibration object's local coordinate system. Again, we adopt the coordinate system associated with the first image as global coordinate frame. The motion for the other two images is given by 3×3 rotation matrices R' and R'' and translation vectors $\mathbf{t}'$ and $\mathbf{t}''$.

With the correct motion estimates, the following points must be collinear:

$$\mathbf{Q} \qquad \begin{pmatrix} \mathsf{R}' & \mathbf{t}' \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \mathbf{Q}' \qquad \begin{pmatrix} \mathsf{R}'' & \mathbf{t}'' \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \mathbf{Q}''$$

We may stack them in the following 4×3 matrix:

$$\begin{pmatrix} Q_1 & R'_{11}Q'_1 + R'_{12}Q'_2 + R'_{13}Q'_3 + t'_1Q'_4 & R''_{11}Q''_1 + R''_{12}Q''_2 + R''_{13}Q''_3 + t''_1Q''_4 \\ Q_2 & R'_{21}Q'_1 + R'_{22}Q'_2 + R'_{23}Q'_3 + t'_2Q'_4 & R''_{21}Q''_1 + R''_{22}Q''_2 + R''_{23}Q''_3 + t''_2Q''_4 \\ Q_3 & R'_{31}Q'_1 + R'_{32}Q'_2 + R'_{33}Q'_3 + t'_3Q'_4 & R''_{31}Q''_1 + R''_{32}Q''_2 + R''_{33}Q''_3 + t''_3Q''_4 \\ Q_4 & Q'_4 & Q''_4 \end{pmatrix} \tag{3}$$

The collinearity constraint means that this matrix must be of rank less than 3, which implies that all sub-determinants of size 3×3 vanish. There are 4 of them, obtained by leaving out one row at a time from the original matrix. Each of these corresponds to a trilinear equation in point coordinates and thus to a trifocal calibration tensor whose coefficients depend on the motion parameters.

Table 4 shows the coefficients of the first two calibration tensors. In both, 34 of the 64 coefficients are always zero. One may observe that the two tensors share several coefficients, e.g. $V_8^1 = V_1^2 = R'_{31}$. The situation is similar for the other two tensors, which are not shown here since the first two are sufficient to compute the motion parameters and thus to perform calibration.

The tensors can be estimated by solving linear equation system, and we verified using simulated random experiments that in general unique solutions (up to scale) are obtained, if 3D points for sufficiently many pixels (29 at least) are available. In the following, we give an

i	C_i	V_i	W_i
1	$Q_1Q'_1Q''_4$	0	R'_{31}
2	$Q_1Q'_2Q''_4$	0	R'_{32}
3	$Q_1Q'_3Q''_4$	0	R'_{33}
4	$Q_1Q'_4Q''_1$	0	$-R''_{31}$
5	$Q_1Q'_4Q''_2$	0	$-R''_{32}$
6	$Q_1Q'_4Q''_3$	0	$-R''_{33}$
7	$Q_1Q'_4Q''_4$	0	$t'_3 - t''_3$
8	$Q_2Q'_1Q''_4$	R'_{31}	0
9	$Q_2Q'_2Q''_4$	R'_{32}	0
10	$Q_2Q'_3Q''_4$	R'_{33}	0
11	$Q_2Q'_4Q''_1$	$-R''_{31}$	0
12	$Q_2Q'_4Q''_2$	$-R''_{32}$	0
13	$Q_2Q'_4Q''_3$	$-R''_{33}$	0
14	$Q_2Q'_4Q''_4$	$t'_3 - t''_3$	0
15	$Q_3Q'_1Q''_4$	$-R'_{21}$	$-R'_{11}$
16	$Q_3Q'_2Q''_4$	$-R'_{22}$	$-R'_{12}$
17	$Q_3Q'_3Q''_4$	$-R'_{23}$	$-R'_{13}$
18	$Q_3Q'_4Q''_1$	R''_{21}	R''_{11}
19	$Q_3Q'_4Q''_2$	R''_{22}	R''_{12}
20	$Q_3Q'_4Q''_3$	R''_{23}	R''_{13}
21	$Q_3Q'_4Q''_4$	$t''_2 - t'_2$	$t''_1 - t'_1$
22	$Q_4Q'_1Q''_1$	$R'_{21}R''_{31} - R''_{21}R'_{31}$	$R'_{11}R''_{31} - R''_{11}R'_{31}$
23	$Q_4Q'_1Q''_2$	$R'_{21}R''_{32} - R''_{22}R'_{31}$	$R'_{11}R''_{32} - R''_{12}R'_{31}$
24	$Q_4Q'_1Q''_3$	$R'_{21}R''_{33} - R''_{23}R'_{31}$	$R'_{11}R''_{33} - R''_{13}R'_{31}$
25	$Q_4Q'_1Q''_4$	$R'_{21}t''_3 - R'_{31}t''_2$	$R'_{11}t''_3 - R'_{31}t''_1$
26	$Q_4Q'_2Q''_1$	$R'_{22}R''_{31} - R''_{21}R'_{32}$	$R'_{12}R''_{31} - R''_{11}R'_{32}$
27	$Q_4Q'_2Q''_2$	$R'_{22}R''_{32} - R''_{22}R'_{32}$	$R'_{12}R''_{32} - R''_{12}R'_{32}$
28	$Q_4Q'_2Q''_3$	$R'_{22}R''_{33} - R''_{23}R'_{32}$	$R'_{12}R''_{33} - R''_{13}R'_{32}$
29	$Q_4Q'_2Q''_4$	$R'_{22}t''_3 - R'_{32}t''_2$	$R'_{12}t''_3 - R_{32}t''_1$
30	$Q_4Q'_3Q''_1$	$R'_{23}R''_{31} - R''_{21}R'_{33}$	$R'_{13}R''_{31} - R''_{11}R'_{33}$
31	$Q_4Q'_3Q''_2$	$R'_{23}R''_{32} - R''_{22}R'_{33}$	$R'_{13}R''_{32} - R''_{12}R'_{33}$
32	$Q_4Q'_3Q''_3$	$R'_{23}R''_{33} - R''_{23}R'_{33}$	$R'_{13}R''_{33} - R''_{13}R'_{33}$
33	$Q_4Q'_3Q''_4$	$R'_{23}t''_3 - R'_{33}t''_2$	$R'_{13}t''_3 - R'_{33}t''_1$
34	$Q_4Q'_4Q''_1$	$R''_{31}t'_2 - R''_{21}t'_3$	$R''_{31}t'_1 - R''_{11}t'_3$
35	$Q_4Q'_4Q''_2$	$R''_{32}t'_2 - R''_{22}t'_3$	$R''_{32}t'_1 - R''_{12}t'_3$
36	$Q_4Q'_4Q''_3$	$R''_{33}t'_2 - R''_{23}t'_3$	$R''_{33}t'_1 - R''_{13}t'_3$
37	$Q_4Q'_4Q''_4$	$t'_2t''_3 - t'_3t''_2$	$t'_1t''_3 - t''_1t'_3$

Table 3: Coupled variables in the trifocal calibration tensors for a general 3D camera. Coefficients not shown here are always zero.

algorithm for computing the motion parameters. Let $V'_i = \lambda V_i$ and $W'_i = \mu W_i, i = 1 \ldots 37$ be the estimated (up to scale) tensors. The algorithm proceeds as follows.

1. Estimate scale factors: $\lambda = \sqrt{V'^2_8 + V'^2_9 + V'^2_{10}}$ and $\mu = \sqrt{W'^2_1 + W'^2_2 + W'^2_3}$.

2. Compute $V_i = \frac{V_i'}{\lambda}$ and $W_i = \frac{W_i'}{\mu}$, $i = 1 \ldots 37$

3. Compute R' and R'':

$$\mathsf{R}' = \begin{pmatrix} -W_{15} & -W_{16} & -W_{17} \\ -V_{15} & -V_{16} & -V_{17} \\ V_8 & V_9 & V_{10} \end{pmatrix} \qquad \mathsf{R}'' = \begin{pmatrix} W_{18} & W_{19} & W_{20} \\ V_{18} & V_{19} & V_{20} \\ -V_{11} & -V_{12} & -V_{13} \end{pmatrix}$$

These matrices will in general not be orthonormal. We "correct" this by computing the orthonormal matrices that are closest to the original matrices (in the sense of the Frobenius norm). To do so, let $\mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$ be the SVD of one of the original matrices. The searched for orthonormal matrix is then given by $\pm\mathsf{U}\mathsf{V}^\mathsf{T}$, where the sign is chosen such that the determinant equals $+1$.

4. Compute $\mathbf{t}'$ and $\mathbf{t}''$ by solving to least squares the following equation system:

$$\underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -R'_{31} & R'_{21} \\ 0 & 0 & 0 & -R'_{31} & 0 & R'_{11} \\ 0 & 0 & 0 & 0 & -R'_{32} & R'_{22} \\ 0 & 0 & 0 & -R'_{32} & 0 & R'_{12} \\ 0 & 0 & 0 & 0 & -R'_{33} & R'_{23} \\ 0 & 0 & 0 & -R'_{33} & 0 & R'_{13} \\ 0 & R''_{31} & -R''_{21} & 0 & 0 & 0 \\ R''_{31} & 0 & -R''_{11} & 0 & 0 & 0 \\ 0 & R''_{32} & -R''_{22} & 0 & 0 & 0 \\ R''_{32} & 0 & -R''_{12} & 0 & 0 & 0 \\ 0 & R''_{33} & -R''_{23} & 0 & 0 & 0 \\ R''_{33} & 0 & -R''_{13} & 0 & 0 & 0 \end{pmatrix}}_{\mathsf{A}} \begin{pmatrix} t'_1 \\ t'_2 \\ t'_3 \\ t''_1 \\ t''_2 \\ t''_3 \end{pmatrix} = \underbrace{\begin{pmatrix} V_{14} \\ V_{21} \\ W_{21} \\ V_{25} \\ W_{25} \\ V_{29} \\ W_{29} \\ V_{33} \\ W_{33} \\ V_{34} \\ W_{34} \\ V_{35} \\ W_{35} \\ V_{36} \\ W_{36} \end{pmatrix}}_{\mathbf{b}}$$

The least squares solution $\left(\mathsf{A}^\mathsf{T}\mathsf{A}\right)^{-1}\mathsf{A}^\mathsf{T}\mathbf{b}$ is well defined since it can be shown that, due to the orthonormality of the rotation matrices R' and R'', the product $\mathsf{A}^\mathsf{T}\mathsf{A}$ is the following invertible matrix:

$$\mathsf{A}^\mathsf{T}\mathsf{A} = \begin{pmatrix} 2 & 0 & 0 & -1 & 0 & 0 \\ 0 & 2 & 0 & 0 & -1 & 0 \\ 0 & 0 & 3 & 0 & 0 & -1 \\ -1 & 0 & 0 & 2 & 0 & 2 \\ 0 & -1 & 0 & 0 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 & 3 \end{pmatrix}$$

Again, using simulations, we verified that this algorithm gives a unique and correct solution in general.

5 Variants of the Calibration Concept

Analogously to the case of 2D cameras, cf. section 3, we developed important specializations of our calibration concept, concerning central cameras and planar calibration objects.

5.1 Central Cameras

In that case, two images are sufficient. Let $\mathbf{C}$ be the optical center (unknown). By proceeding as above, we obtain 4 bifocal calibration tensors of size 4×4 and rank 2, that are somewhat similar to fundamental matrices. One of them is shown here:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ R'_{31}C_4 & R'_{32}C_4 & R'_{33}C_4 & -C_3 + C_4 t'_3 \\ -R'_{21}C_4 & -R'_{22}C_4 & -R'_{23}C_4 & C_2 - C_4 t'_2 \\ R'_{21}C_3 - R'_{31}C_2 & R'_{22}C_3 - R'_{32}C_2 & R'_{23}C_3 - R'_{33}C_2 & C_3 t'_2 - C_2 t'_3 \end{pmatrix}$$

Due to lack of space, we only describe parts of these results. It is relatively straightforward to extract the motion parameters and the optical center from these tensors.

5.2 General 3D Cameras with a Planar Calibration Object

We make the following change to incorporate the planarity constraint.

$$Q_3 = Q'_3 = Q''_3 = 0$$

We again form the 4×3 matrix with the triplet in one common coordinate frame as follows.

$$\begin{pmatrix} Q_1 & R'_{11}Q'_1 + R'_{12}Q'_2 + t'_1 Q'_4 & R''_{11}Q''_1 + R''_{12}Q''_2 + t''_1 Q''_4 \\ Q_2 & R'_{21}Q'_1 + R'_{22}Q'_2 + t'_2 Q'_4 & R''_{21}Q''_1 + R''_{22}Q''_2 + t''_2 Q''_4 \\ 0 & R'_{31}Q'_1 + R'_{32}Q'_2 + t'_3 Q'_4 & R''_{31}Q''_1 + R''_{32}Q''_2 + t''_3 Q''_4 \\ Q_3 & Q'_4 & Q''_4 \end{pmatrix}$$

We obtain four tensors by removing one row at a time and expanding the matrix. Using simulations, we found that the removal of third row does not provide unique solution. The reason might be that, while removing the third row, we do not actually utilize the planarity constraint. However we use only the first two tensor equations for computing the motion parameters.

The extraction of individual motion parameters is slightly complicated compared to the earlier cases. First of all we do not have any constraint to compute the scale parameter. We compute V and W up to a common scale λ, since they share some common variables.

$$\begin{aligned} \lambda R'_{3,1} &= V'_6 & \lambda R'_{3,2} &= V'_7 \\ \lambda R''_{3,1} &= -V'_8 & \lambda R''_{3,2} &= -V'_9 \\ scaled(t'_3) = \lambda t'_3 &= \frac{(-V'_9 V'_{17} + V'_8 V'_{18}) V'_6}{-V'_9 V'_{11} + V'_8 V'_{12}} \\ scaled(t''_3) = \lambda t''_3 &= (\lambda t'_3) - V'_{10} \end{aligned}$$

For extracting the remaining parameters we follow an indirect approach.

i	C_i	V_i	W_i
1	$Q_1Q'_1Q''_4$	0	$R'_{3,1}$
2	$Q_1Q'_2Q''_4$	0	$R'_{3,2}$
3	$Q_1Q'_4Q''_1$	0	$-R''_{3,1}$
4	$Q_1Q'_4Q''_2$	0	$-R''_{3,2}$
5	$Q_1Q'_4Q''_4$	0	$t'_3 - t''_3$
6	$Q_2Q'_1Q''_4$	$R'_{3,1}$	0
7	$Q_2Q'_2Q''_4$	$R'_{3,2}$	0
8	$Q_2Q'_4Q''_1$	$-R''_{3,1}$	0
9	$Q_2Q'_4Q''_2$	$-R''_{3,2}$	0
10	$Q_2Q'_4Q''_4$	$t'_3 - t''_3$	0
11	$Q_4Q'_1Q''_1$	$-R''_{2,1}R'_{3,1} + R'_{2,1}R''_{3,1}$	$R'_{1,1}R''_{3,1} - R''_{1,1}R'_{3,1}$
12	$Q_4Q'_1Q''_2$	$R'_{2,1}R''_{3,2} - R''_{2,2}R'_{3,1}$	$R'_{1,1}R''_{3,2} - R''_{1,2}R'_{3,1}$
13	$Q_4Q'_1Q''_4$	$R'_{2,1}t''_3 - t''_2R'_{3,1}$	$R'_{1,1}t''_3 - t''_1R'_{3,1}$
14	$Q_4Q'_2Q''_1$	$-R''_{2,1}R'_{3,2} - R''_{2,2}R'_{3,2}$	$R'_{1,2}R''_{3,1} - R''_{1,1}R'_{3,2}$
15	$Q_4Q'_2Q''_2$	$R'_{2,2}R''_{3,2} - R''_{2,2}R'_{3,2}$	$R'_{1,2}R''_{3,2} - R''_{1,2}R'_{3,2}$
16	$Q_4Q'_2Q''_4$	$R'_{2,2}t''_3 - t''_2R'_{3,2}$	$R'_{1,2}t''_3 - t''_1R'_{3,2}$
17	$Q_4Q'_4Q''_1$	$-R''_{2,1}t'_3 + t'_2R''_{3,1}$	$t'_1R''_{3,1} - R''_{1,1}t'_{3,1}$
18	$Q_4Q'_4Q''_2$	$t'_2R''_{3,2} - R''_{2,2}t'_3$	$t'_1R''_{3,2} - R''_{1,2}t'_{3,1}$
19	$Q_4Q'_4Q''_4$	$t'_2t''_3 - t''_2t'_3$	$t'_1t''_3 - t''_1t'_3$

Table 4: Coupled variables in the 3D general camera with a planar board pattern

$$\begin{pmatrix} -V'_8 & 0 & -V'_6 & 0 \\ -V'_9 & 0 & 0 & -V'_6 \\ 0 & -V'_8 & -V'_7 & 0 \\ 0 & -V'_9 & 0 & -V'_7 \end{pmatrix} \begin{pmatrix} R'_{2,1} \\ R'_{2,2} \\ R''_{2,1} \\ R''_{2,2} \end{pmatrix} = \begin{pmatrix} V'_{11} \\ V'_{12} \\ V'_{14} \\ V'_{15} \end{pmatrix}$$

The rank of the above system is 3 and hence we obtain the solution in the subspace spanned by two vectors.

$$\begin{pmatrix} R'_{2,1} \\ R'_{2,2} \\ R''_{2,1} \\ R''_{2,2} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} + l_1 \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

Similarly for extracting $R'_{1,1}$,$R'_{1,2}$,$R''_{1,1}$ and $R''_{1,2}$, we form the following linear system.

$$\begin{pmatrix} -V'_8 & 0 & -V'_6 & 0 \\ -V'_9 & 0 & 0 & -V'_6 \\ 0 & -V'_8 & -V'_7 & 0 \\ 0 & -V'_9 & 0 & -V'_7 \end{pmatrix} \begin{pmatrix} R'_{1,1} \\ R'_{1,2} \\ R''_{1,1} \\ R''_{1,2} \end{pmatrix} = \begin{pmatrix} W'_{11} \\ W'_{12} \\ W'_{14} \\ W'_{15} \end{pmatrix}$$

As in the previous case, the rank of the above system is 3.

$$\begin{pmatrix} R'_{1,1} \\ R'_{1,2} \\ R''_{1,1} \\ R''_{1,2} \end{pmatrix} = \begin{pmatrix} a_5 \\ a_6 \\ a_7 \\ a_8 \end{pmatrix} + l_1 \begin{pmatrix} b_5 \\ b_6 \\ b_7 \\ b_8 \end{pmatrix}$$

We estimate the values of l_1 and l_2 using orthonormal properties of the rotation matrix.

$$\begin{aligned} R'_{1,1}R'_{1,2} + R'_{2,1}R'_{2,2} + R'_{3,1}R'_{3,2} &= 0 \\ R''_{1,1}R''_{1,2} + R''_{2,1}R''_{2,2} + R''_{3,1}R''_{3,2} &= 0 \\ R'^2_{1,1} + R'^2_{2,1} + R'^2_{3,1} &= 1 \\ R'^2_{1,2} + R'^2_{2,2} + R'^2_{3,2} &= 1 \\ R''^2_{1,1} + R''^2_{2,1} + R''^2_{3,1} &= 1 \\ R''^2_{1,2} + R''^2_{2,2} + R''^2_{3,2} &= 1 \end{aligned}$$

On substituting the rotation variables and simplifying the expression we get the following system.

$$\begin{pmatrix} a_1b_2 + b_1a_2 & b_1b_2 & a_5b_6 + b_5a_6 & b_5b_6 & V'_6V'_7 \\ a_3b_4 + b_3a_4 & b_3b_4 & a_7b_8 + b_8a_7 & b_7b_8 & V'_8V'_9 \\ 2a_1b_1 & b_1^2 & 2a_5b_5 & b_5^2 & V'^2_6 \\ 2a_2b_2 & b_2^2 & 2a_6b_6 & b_6^2 & V'^2_7 \\ 2a_3b_3 & b_3^2 & 2a_7b_7 & b_7^2 & V'^2_8 \\ 2a_4b_4 & b_4^2 & 2a_8b_8 & b_8^2 & V'^2_9 \end{pmatrix} \begin{pmatrix} l_1 \\ l_1^2 \\ l_2 \\ l_2^2 \\ \frac{1}{\lambda^2} \end{pmatrix} = \begin{pmatrix} -a_5a_6 - a_1a_2 \\ -a_7a_8 - a_3a_4 \\ 1 - a_1^2 - a_5^2 \\ 1 - a_2^2 - a_6^2 \\ 1 - a_3^2 - a_7^2 \\ 1 - a_4^2 - a_8^2 \end{pmatrix}$$

The rank of the above system is 3 and we obtain solutions for l_1,l_2, l_1^2,l_2^2 and $\frac{1}{\lambda^2}$ in a subspace spanned by three vectors. Next we use the constraints between l_1,l_2,l_1^2 and l_2^2 to obtain unique solutions for l_1 and l_2. Finally we use the values of l_1 and l_2 to compute $R'_{1,1}$, $R'_{1,2}$, $R'_{2,1}$,$R'_{2,2}$,$R''_{1,1}$, $R''_{1,2}$,$R''_{2,1}$ and $R''_{2,2}$.

$$\lambda = \pm\sqrt{\frac{-R'_{3,1}R'_{3,2}}{R'_{1,1}R'_{1,2} + R'_{2,1}R'_{2,2}}}$$

$$R'_{3,1} = \frac{V'_6}{\lambda} \qquad R'_{3,2} = \frac{V'_7}{\lambda}$$

$$R''_{3,1} = \frac{-V'_8}{\lambda} \qquad R''_{3,2} = \frac{-V'_9}{\lambda}$$

$$t'_1 = \frac{W'_{17} + R''_{1,1}scaled(t'_3)}{-V'_8} \qquad t'_2 = \frac{V'_{17} + R''_{3,1}scaled(t'_3)}{-V'_8} \qquad t'_3 = \frac{scaled(t'_3)}{\lambda}$$

$$t''_1 = \frac{R'_{1,1}scaled(t''_3) - W'_{13}}{V'_6} \qquad t''_2 = \frac{R'_{2,1}scaled(t''_3) - V'_{13}}{V'_6} \qquad t''_3 = \frac{scaled(t''_3)}{\lambda}$$

i	j	k	V_i	W_i	M_i	N_i
1	1	1	0	$R'_{3,1}$	0	$R''_{3,1}$
2	1	2	0	$R'_{3,2}$	0	$R''_{3,1}$
3	1	4	0	$-O_3 + t'_3$	0	$-O_3 + t''_3$
4	2	1	$R'_{3,1}$	0	$R''_{3,1}$	0
5	2	2	$R'_{3,2}$	0	$R''_{3,2}$	0
6	2	4	$-O_3 + t'_3$	0	$-O_3 + t''_3$	0
7	4	1	$-O_2R'_{3,1} + O_3R'_{2,1}$	$-O_1R'_{3,1} + O_3R'_{1,1}$	$-O_2R''_{3,1} + O_3R''_{2,1}$	$-O_1R''_{3,1} + O_3R''_{1,1}$
8	4	2	$-O_2R'_{3,2} + O_3R'_{2,2}$	$-O_1R'_{3,2} + O_3R'_{1,2}$	$-O_2R''_{3,2} + O_3R''_{2,2}$	$-O_1R''_{3,2} + O_3R''_{1,2}$
9	4	4	$-O_2t'_3 + O_3t'_2$	$-O_1t'_3 + O_3t'_1$	$-O_2t''_3 + O_3t''_2$	$-O_1t''_3 + O_3t''_1$

Table 5: Coupled variables in four of the bifocal matching tensors in a 3D single center camera with a planar calibration grid.

5.3 Central Cameras with Planar Calibration Objects

Earlier we have seen some cases with central cameras using two views. In this case however, the associated calibration tensors do not contain sufficient information in order to uniquely estimate the motion and optical center (O). This is actually not surprising: even in the more restricted case of **perspective** cameras with 5 intrinsic parameters, two views of a planar calibration object do not suffice [19, 23]. We thus developed an algorithm that uses three views. It is rather complicated, too, but was shown to provide unique solutions in general. On stacking the collinear points we get the following 4×4 matrix

$$\begin{pmatrix} O_1 & Q_1 & R'_{11}Q'_1 + R'_{12}Q'_2 + t'_1Q'_4 & R''_{11}Q''_1 + R''_{12}Q''_2 + t''_1Q''_3 \\ O_2 & Q_2 & R'_{21}Q'_1 + R'_{22}Q'_2 + t'_2Q'_3 & R''_{21}Q''_1 + R''_{22}Q''_2 + t''_2Q''_3 \\ O_3 & 0 & R'_{31}Q'_1 + R'_{32}Q'_2 + t'_3Q'_3 & R''_{31}Q''_1 + R''_{32}Q''_2 + t''_3Q''_3 \\ 1 & Q_4 & Q'_4 & Q''_4 \end{pmatrix}$$

Here we get 16 constraints by removing one row and one column at a time. However we obtained unique solutions for only 6 cases, out of which we use 4 cases to extract the motion variables. To distinguish each constraint in the set of 16 constraints we use $T_{i,j}$ to refer to the constraint obtained by removing i_{th} row and j_{th} column. The constraints corresponding to $T_{4,1}, T_{4,2}, T_{3,1}$ and $T_{3,2}$ are $\sum_{i=1}^{i=9} C4_iV_i = 0$, $\sum_{i=1}^{i=9} C4_iW_i = 0$, $\sum_{i=1}^{i=9} C3_iM_i = 0$ and $\sum_{i=1}^{i=9} C3_iN_i$ respectively, where $C4_i = Q_jQ'_k$,$C3_i = Q_jQ''_k$ and the other variables are given in Table 5. We do not have any direct methods to compute the scale factor. We first compute some rotation components up to a scale as shown below.

$$\lambda_1 R'_{3,1} = V_4 \qquad \lambda_1 R'_{3,2} = V_5 \qquad \lambda_2 R''_{3,1} = M_4 \qquad \lambda_2 R''_{3,2} = M_5$$

We obtain the following equations by multiplying certain variables in Table 5.

$$\begin{pmatrix} 0 & -V_4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -V_5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -V_6' & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -V_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -V_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -V_6' & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -M_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -M_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -M_6' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -M_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -M_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -N_6' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} O_1 \\ O_2 \\ \lambda_1 O_3(t_1' - O_1) \\ \lambda_1 O_3(t_2' - O_2) \\ \lambda_1 O_3 R_{1,1}' \\ \lambda_1 O_3 R_{1,2}' \\ \lambda_1 O_3 R_{2,1}' \\ \lambda_1 O_3 R_{2,2}' \\ \lambda_2 O_3(t_1'' - O_1) \\ \lambda_2 O_3(t_2'' - O_2) \\ \lambda_2 O_3 R_{1,1}'' \\ \lambda_2 O_3 R_{1,1}'' \\ \lambda_2 O_3 R_{2,1}'' \\ \lambda_2 O_3 R_{2,2}'' \end{pmatrix} = \begin{pmatrix} V_7' \\ V_8' \\ V_9' \\ W_7' \\ W_8' \\ W_9' \\ M_7' \\ M_8' \\ M_9' \\ N_7' \\ N_8' \\ M_9' \end{pmatrix}$$

The above set of equations is of rank 12. By coupling certain variables, we reduce the number of variables to 14 and extract them in a subspace spanned by a set of three vectors as shown below.

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \\ u_{13} \\ u_{14} \end{pmatrix} = \begin{pmatrix} O_1 \\ O_2 \\ \lambda_1 O_3(t_1' - O_1) \\ \lambda_1 O_3(t_2' - O_2) \\ \lambda_1 O_3 R_{1,1}' \\ \lambda_1 O_3 R_{1,2}' \\ \lambda_1 O_3 R_{2,1}' \\ \lambda_1 O_3 R_{2,2}' \\ \lambda_2 O_3(t_1'' - O_1) \\ \lambda_2 O_3(t_2'' - O_2) \\ \lambda_2 O_3 R_{1,1}'' \\ \lambda_2 O_3 R_{1,1}'' \\ \lambda_2 O_3 R_{2,1}'' \\ \lambda_2 O_3 R_{2,2}'' \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \end{pmatrix} + l_1 \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \\ b_{10} \\ b_{11} \\ b_{12} \\ b_{13} \\ b_{14} \end{pmatrix} + {}_2 \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \\ c_{10} \\ c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \end{pmatrix}$$

We now use the constraints in the rotational matrix to evaluate the values of l_1 and l_2.

$$\begin{aligned} u_5 u_6 + u_7 u_8 + V_4 V_5 O_3^2 &= 0 \\ u_{11} u_{12} + u_{13} u_{14} + M_4 M_5 O_3^2 &= 0 \\ u_5^2 + u_7^2 + (V_4 O_3)^2 &= \lambda_1^2 \\ u_6^2 + u_8^2 + (V_4 O_3)^2 &= \lambda_1^2 \\ u_{11}^2 + u_{13}^2 + (M_4 O_3)^2 &= \lambda_2^2 \\ u_{12}^2 + u_{14}^2 + (M_5 O_3)^2 &= \lambda_2^2 \end{aligned}$$

By further simplifying we get the following equations in terms of l_1,l_2,$l_1 l_2$,l_1^2,l_2^2,O_3^2, $\frac{1}{\lambda_1 O_3^2}$ and $\frac{1}{\lambda_2 O_3^2}$. The system of equations has 8 variables in 6 equations. The rank of this system of

equations was 5 and thus we obtained solution for these 8 variables in the subspace spanned by three vectors. Then we used the relationship between the variables to compute the exact solution for l_1 and l_2. Next we estimate the 14 variables(u_i's), which depend on l_1 and l_2.

$$O_3 = \pm\sqrt{\frac{u_5u_6 + u_7u_8}{-V_4V_5}}$$

$$\lambda_1 = \pm\sqrt{u_5^2 + u_7^2 + (V_4O_3)^2}$$

$$\lambda_2 = \pm\sqrt{u_{11}^2 + u_{13}^2 + (M_4O_3)^2}$$

$$R'_{1,1} = \frac{u_5}{O_3\lambda_1} \qquad R'_{1,2} = \frac{u_6}{O_3\lambda_1}$$

$$R'_{2,1} = \frac{u_7}{O_3\lambda_1} \qquad R'_{2,2} = \frac{u_8}{O_3\lambda_1}$$

$$R''_{1,1} = \frac{u_{11}}{O_3\lambda_2} \qquad R''_{1,2} = \frac{u_{12}}{O_3\lambda_2}$$

$$R''_{2,1} = \frac{u_{13}}{O_3\lambda_2} \qquad R''_{2,2} = \frac{u_{14}}{O_3\lambda_2}$$

$$R'_{3,1} = \frac{V_4}{\lambda_1} \qquad R'_{3,2} = \frac{V_5}{\lambda_1}$$

$$R''_{3,1} = \frac{M_4}{\lambda_2} \qquad R''_{3,2} = \frac{M_5}{\lambda_2}$$

$$t'_1 = \frac{u_3 + O_1O_2\lambda_1}{O_3\lambda_1} \qquad t'_2 = \frac{u_4 + O_2O_3\lambda_1}{O_3\lambda_1} \qquad t'_3 = \frac{O_3t'_2 - V_9}{O_2}$$

$$t''_1 = \frac{u_9 + O_1O_2\lambda_2}{O_3\lambda_2} \qquad t''_2 = \frac{u_{10} + O_2O_3\lambda_2}{O_3\lambda_2} \qquad t'_3 = \frac{O_3t''_2 - M_9}{O_2}$$

6 Experimental Evaluation

As mentioned above, we verified each algorithm using simulated random experiments, cf. figure 3. This was first done using noiseless data. We also tested our methods using noisy data and obtained satisfying results. A detailed quantitative analysis remains yet to be carried out.

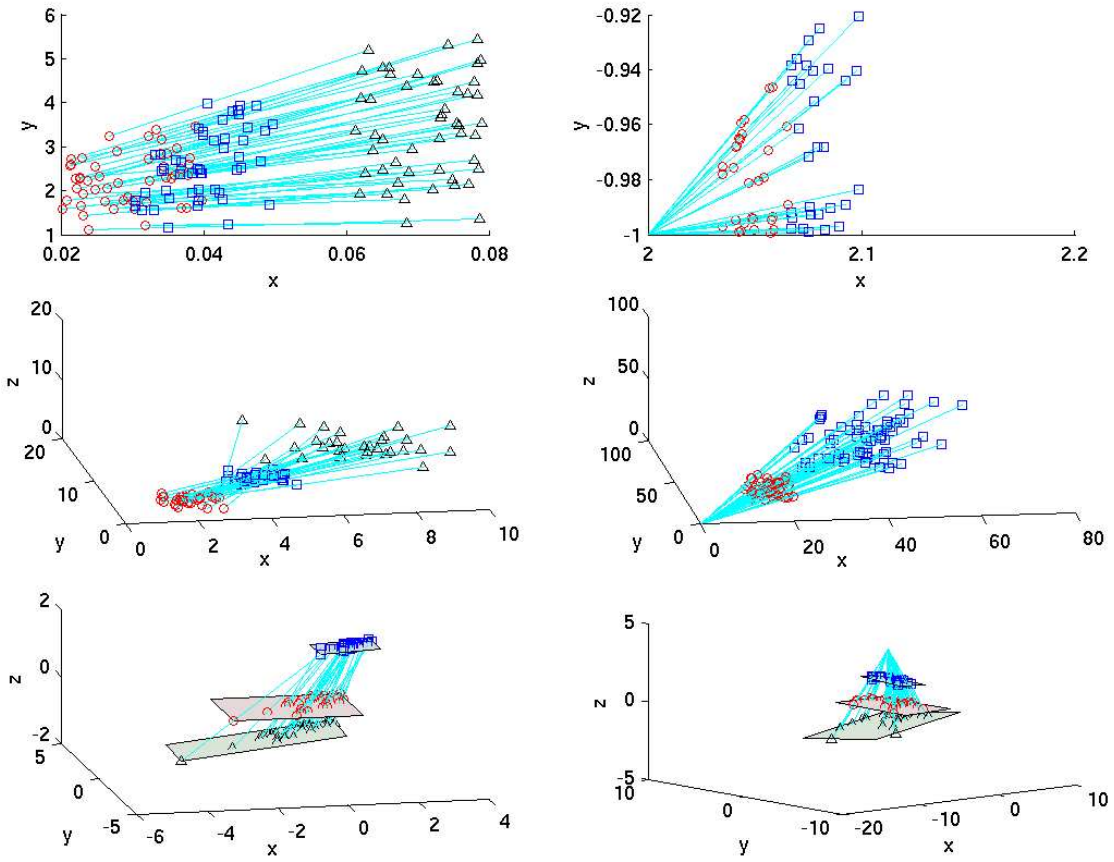


Figure 3: Simulation data for 30 rays are shown for 2D general camera, 2D central camera, 3D general camera, 3D central camera, 3D general camera with planar calibration object and finally 3D central camera with planar calibration object. The red circles, blue squares and the black triangles show the points in first, second and third views respectively.

We did various experiments with real images, using a 3D-Pixel digital camera with moderate optical distortions, a camera with a fish-eye lens and "home-made" catadioptric systems consisting of a digital camera and various curved off-the-shelf mirrors. We used planar calibration objects consisting of black dots or squares on white paper. Figure 4 shows the three board position for a pinhole camera and a fish eye lens.

We use Harris corner detection to extract the dots and later process the nearby pixels to compute the center of the dots up to a subpixel level accuracy. For every image pixel corresponding to every dot in the first board, we compute the 3D positions in the second and the third boards using 4-point homography for the neighbouring pixels, cf figure 5. It is very important to note that the four neighbouring dots which are selected should not have any subset of 3 collinear dots. The homography based computation makes a continuity assumption in the image in contrast to our goal of generic calibration approach. This may not be very accurate

for certain cameras, however other mechanisms to solve this problem come with a compromise on the practicality of our technique.

Also it is very difficult to get the 3D positions for every image pixel in all the 3 boards. Precisely speaking, we can calibrate only the image region common (intersection of convex-hulls of dots in the three views) to all the three boards. However in the case of central cameras, after computing the center using the common region, we can compute the rays for other pixels too.

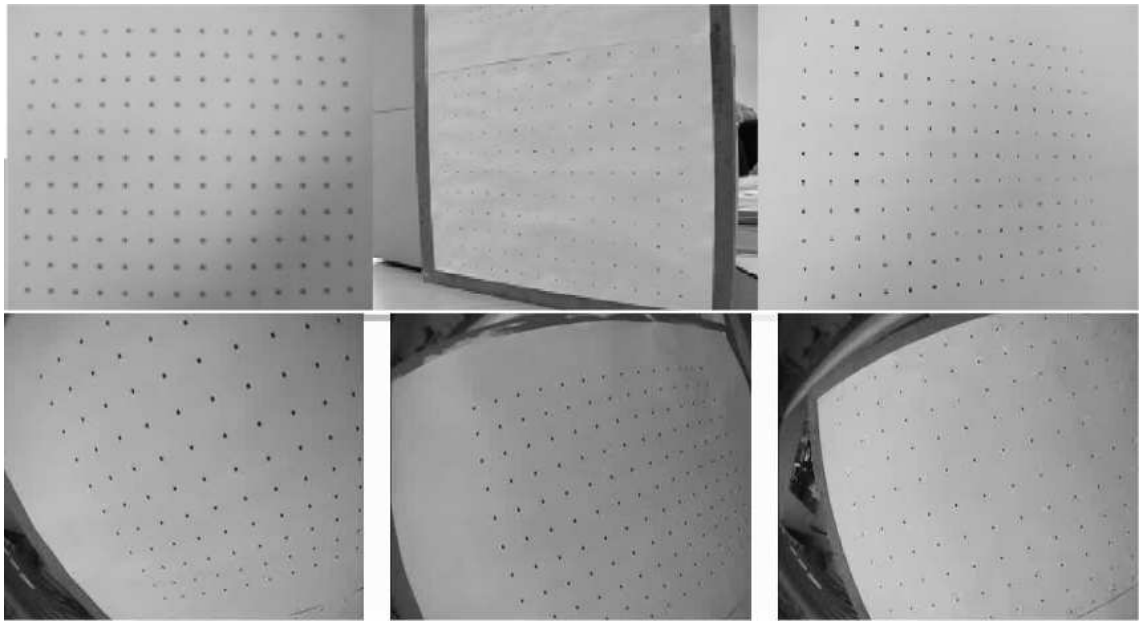


Figure 4: Images of three boards of different sizes, captured by a pinhole camera and a fish eye lens, are shown in top and bottom respectively.

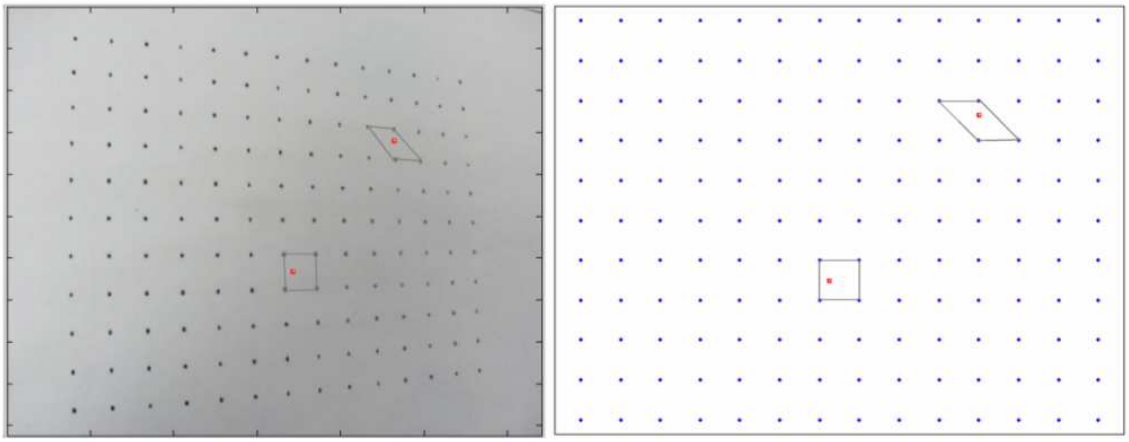
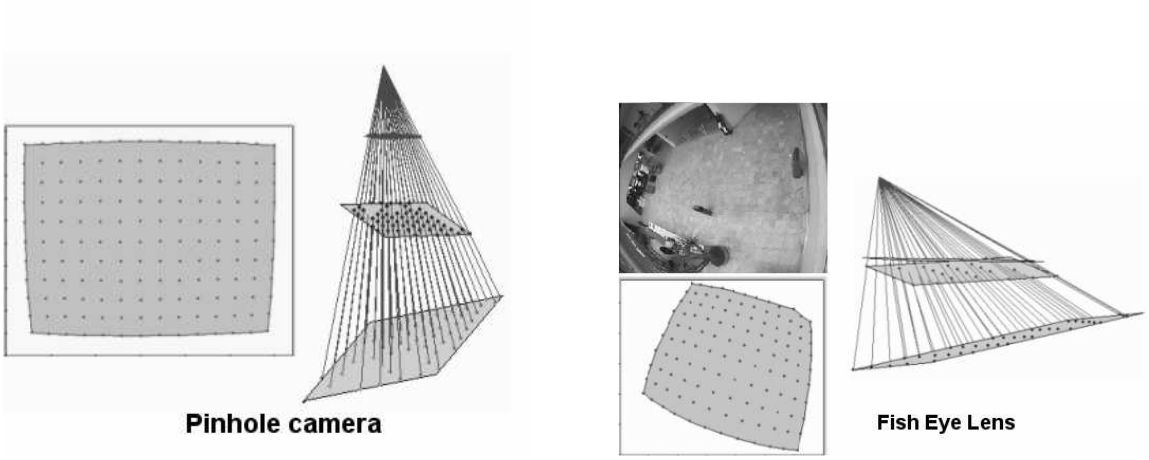


Figure 5: On the left we show the image of the calibration object in the second view. On the right we show the 3D coordinates of the dots. We see two pixels on the left, which do not lie on the dots, and thus do not have known 3D locations. We use the 4-point homography for the nearby dots to obtain their 3D locations.

6.1 Pinhole and Fish-Eye Cameras

On applying the theory developed for single center planar board camera we obtained the results as shown in Figure 6. You can observe that we have calibrated most part of the image with just three boards. The calibrated camera rays and the pose of the calibration objects, which is given by the estimated motion parameters, are shown. It is difficult to evaluate the calibration quantitatively, but we observe that for every pixel considered, the estimated motion parameters give rise to nearly perfectly collinear calibration points. By analysing of singular values for the tensors, we found that by using boards of three different sizes we obtained stable and unique solutions. The board positions can be non intersecting, well aligned to each other and still have a large common area of interesection. We are also working on increasing the number of boards and using nonlinear refinement technique for complete calibration.



(a) (b)

Figure 6: The shaded region shows the calibrated region and the rays in 3D space passing through the three boards, in the process of calibration, are shown for pinhole and fish eye lens.

6.2 Home-Made Non-Central Catadioptric Cameras

We also consider some catadioptric home-made non-central cameras to test our calibration method, given in section 5.2. From the discussion in [2], we know that only a certain catadioptric systems are central. Clearly the catadioptric home-made configurations we consider in Figure 7 are all non-central.

Figure 7: Different combinations of mirrors, lenses, boards and the pinhole camera to generate different cameras are shown.

7 Applications

We have considered 3 different applications to evaluate our calibration approach.

- Perspective view synthesis
- structure and motion recovery
- epipolar curve analysis

7.1 Persective View Synthesis

At the end of the calibration for a central camera, we obtain a bunch of rays passing passing through the center, each associated with a specific color from its image pixel. However the rays need not be perspective. In other words, we mean that a line in 3D space may not remain a line in the image. To obtain a persective view we first intersect the bunch of rays with a plane on which we want the persective view. We give the color of the ray to its point of interesection

on the plane. Finally we compute the bounding box for the points of intersection on the plane and use interpolation or nearest neighbour color for the rest of the points inside the bounding box. We sliced the rays from a pinhole camera and a fish eye lens for two scenes (MOVI house and a different board other than the calibration boards) for generating the perspective views as shown in Figure 8.

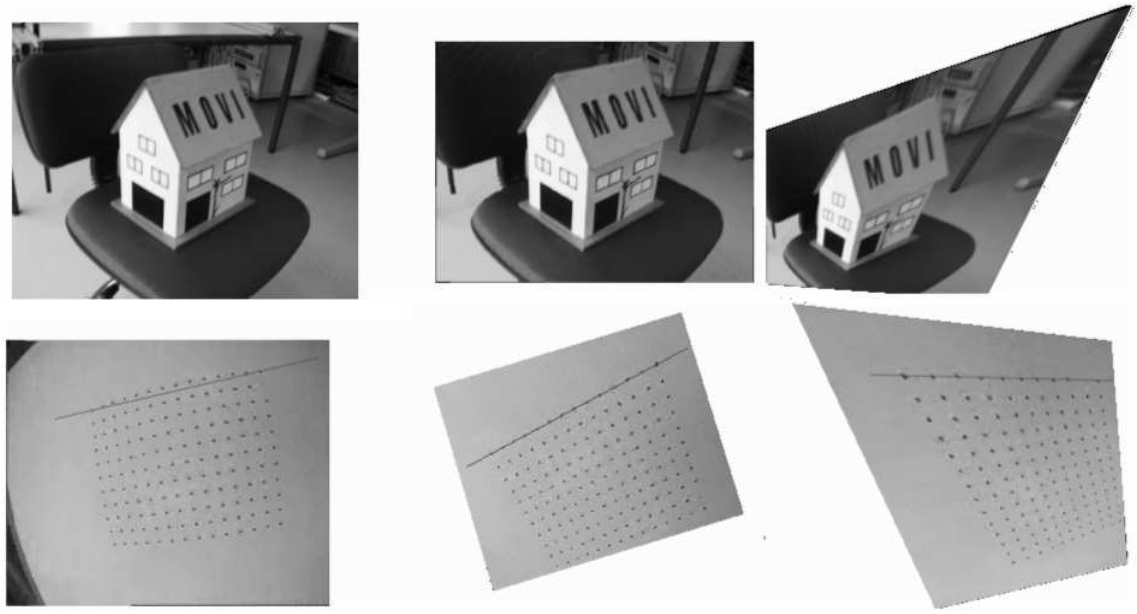


Figure 8: In the top, we see a pinhole image and its two perspective generated images for the calibrated region. An image taken by a fish eye lens and its two perspective images are shown in the bottom.

7.2 Motion Recovery

After obtaining the image correspondences, we compute the motion by applying the constraint that the corresponding 3D rays intersect (see Figure 9). O and O' are the camera centers for the first and the second views. Let us consider a point X, which is seen by both the views. The point at infinity along the ray OX is a. Let a' be the point at infinity in a different coordinate system. Let the motion vector from the second view to the first view be given by (R,t). Let D be the point at infinity along the ray OO'. Since a point at infinity is not affected by translation Ra' lies along the ray O'X. Now we know that a,Ra' and D lie on a single line, which is a line at infinity.

$$a'^T \underbrace{R^T [D]_\times}_{E} a = 0$$

We need 8 correspondences to extract the motion parameters from the 3×3 matrix E, in the same way as we extract the motion parameters from the essential matrix.

7.3 Epipolar Curves

For computing the epipolar curve in the second view for a pixel x in the first view, we first compute the rays in the second view which intersect the ray corresponding to x. Image pixels

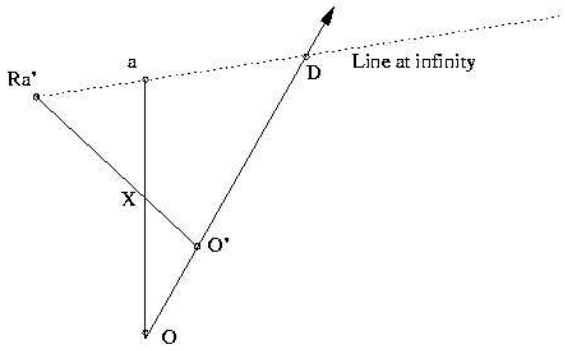


Figure 9: Motion Recovery.

of these intersecting rays form the epipolar curve. In Figure 10 we show the epipolar curves for pinhole and fisheye cameras.

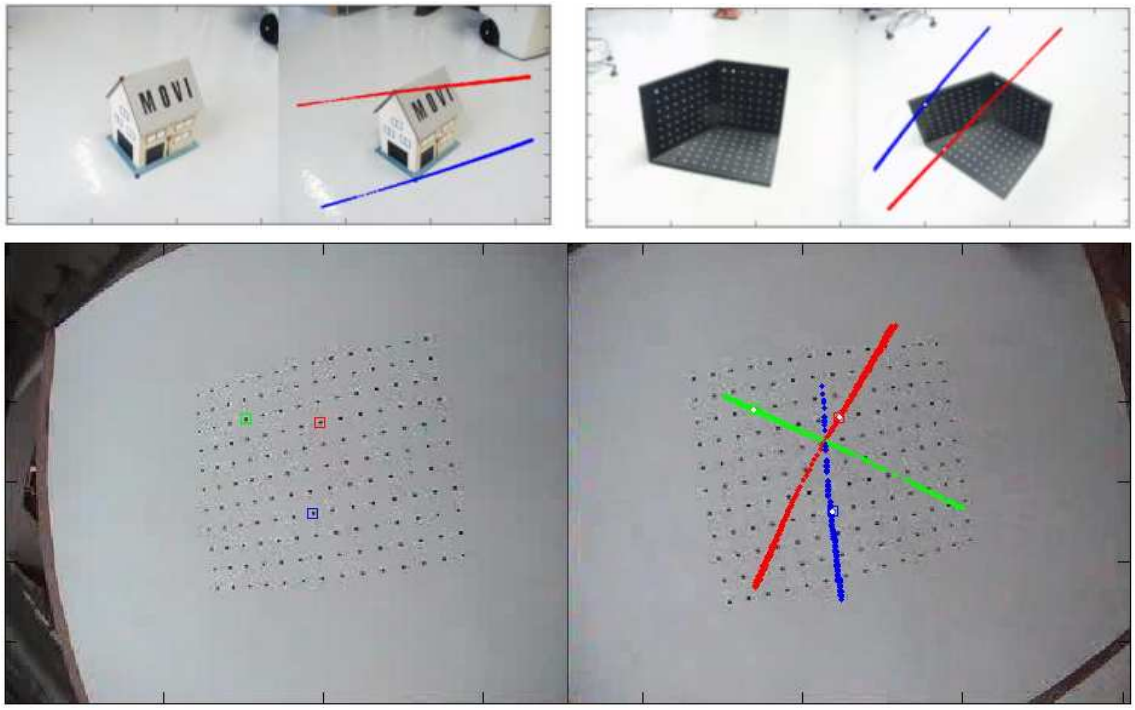
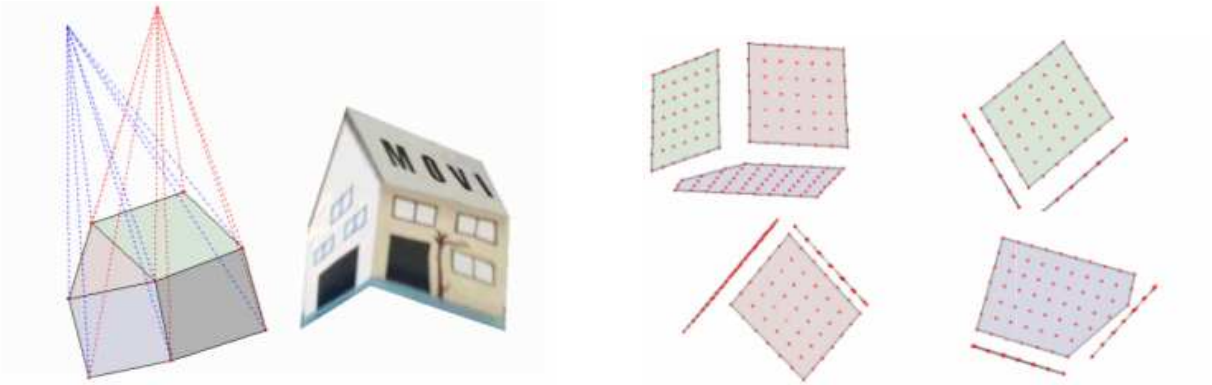


Figure 10: Top:Epipolar curves(for 2 points) for two scenes in pinhole images. Bottom:Epipolar curves (for 3 points) for fisheye images. These are not straight lines, but intersect in a single point, since we use a central camera.

7.4 Structure Recovery

After computing the image correspondences, we intersect the rays corresponding to matching pixels to compute its 3D point in space. Some of the results of this 3D reconstruction algorithm using ray intersection for images are shown in Figure 11. We found the results to be very accurate. The angle between the right angled faces in the house is estimated to be 89.5 without doing any nonlinear refinement. In the three plane calibration pattern, we found that the

reconstruction points lie very accurately on the plane and the angle between the planes (see Figure 11) also matched well with the ground truth. One can also infer that this way of calibration enables you to obtain the motion and structure for any kind of camera (single center cameras, nonsingle center cameras, cameras with distortions, etc) without any distinction.



(a) (b)

Figure 11: We show the results of the ray intersection technique for obtaining the 3D reconstruction for two scenes, a)house and b)calibration pattern. The images used in the reconstruction are shown in Figure 10

8 Discussion

The algorithm for central cameras seems to work fine, even with minimum input of 3 views and a planar calibration object. Experiments with non-central catadioptric cameras however did so far not give satisfying results. One reason for poor stability of the non-central method is the way we currently obtain our input (homography-based interpolation of calibration points). We also think that the general algorithm, which is essentially based on solving linear equations, can only give stable results with minimum input (3 views) if the considered camera is clearly non-central. By this, we mean that there is not any point that is "close" to all camera rays, which might e.g. be the case for stereo systems[2].

We propose several ideas for overcoming these problems. Most importantly, we probably need to use several to many images for a stable calibration. We have ideas for a bundle adjustment formulation of the calibration problem, which is not straightforward, since the camera model is of discrete nature; e.g. it does not give a direct way of handling projection of 3D points to sub-pixel image coordinates. For initialization of the non-central bundle adjustment, we may use the (stabler) calibration results for the central model. Another way of stabilizing the calibration might be the possible inclusion of constraints on the set of camera rays, such as rotational or planar symmetry, if appropriate.

Although we now have a single algorithm that works for nearly all existing camera types, different cameras will likely require different designs of calibration objects, e.g. panoramic

[2]We actually tested the general algorithm with a stereo rig consisting of two perspective cameras. The result was quite stable, but we detected a generic 1-dof-ambiguity in the calibration solution for this special case, that has to be further investigated.

cameras vs. narrow-field-of-view ones. In our experiments, we already use planar calibration objects of different sizes for the different views, cf. figure 4. This way, we can place them such that they do not "intersect" in space, which would give less stable results, especially for the camera rays passing close to the intersection region. We also plan to use different calibration objects for initialization and bundle adjustment: initialization, at least for the central model, can be performed using the type of calibration object used in this work. As for bundle adjustment, we might then switch to objects with a much denser "pattern" e.g. with a coating consisting of randomly distributed colored speckles. Another possibility is to use a flat screen to produce a dense set of calibration points [7].

9 Conclusions

We have proposed a theory and algorithms for a highly general calibration concept. We consider this as for now mainly as a conceptual contribution: it shows how to calibrate (nearly) any camera, using one and the same algorithm. Further, the approach was specialized to the useful special case of central cameras, and algorithms for the use of planar calibration objects, were developed too. We demonstrated by preliminary experiments that the approach allows to calibrate central cameras without using any distortion model, with applications in distortion correction and motion estimation. Other standard vision tasks, such as pose estimation and 3D reconstruction, are straightforward to formulate and to carry out for the considered general imaging model.

We believe in the concept's potential for calibrating cameras with "exotic" distortions – such as fish-eye lenses with hemispheric field of view or catadioptric cameras, especially non-central ones. To this end, we need to make the method stabler, e.g. by extending it such that it can use many images and by designing better calibration objects.

A Analysis of Underconstrained Cases for 2D Cameras

Here, we investigate what happens when we use the general algorithm, but with data that stems from a central camera. In that case, the linear estimation of our tensors does not give a unique solution, and the subsequent calibration algorithm will fail. In the following, we analyze the degree of ambiguity of the solution. Our main motivation for doing this analysis is related to the calibration of stereo system: a stereo system consisting of central cameras, is considered here as a single non-central sensor. It turns out unhappily that a stereo system consisting of 2 central cameras, does not lead to a unique solution of the general calibration method. It is shown below that the minimum case of a stereo system that can be calibrated using the basic algorithm for non-central cameras, is a system with 3 central cameras.

To carry out our analysis, we simulate data points that are already aligned, i.e. the true solution for the motions is the identity. We then check if there exist other solutions, which would mean that calibration using the general method fails.

We first deal with the case of a single central 2D camera, for which no unique solution exists. Then, it is shown that a system with two central 2D cameras, already can be calibrated.

Then, we consider the 3D case, for one, two and three central cameras.

A.1 A Single Central 2D Camera

Points are already aligned, so each triplet of correspondences can be modeled by variables X and Y (for the ray on which they lie) and scales s, s', s'', that express the positions of the points along the ray:

$$\mathbf{Q} = \begin{pmatrix} X \\ Y \\ s \end{pmatrix} \qquad \mathbf{Q}' = \begin{pmatrix} X \\ Y \\ s' \end{pmatrix} \qquad \mathbf{Q}'' = \begin{pmatrix} X \\ Y \\ s'' \end{pmatrix}$$

We plug these coordinates into equation (1) and the associated table 1 of its coefficients. This gives us table 6.

i	C_i	V_i
1	$s''(X^2+Y^2)$	R'_{21}
2	0	R'_{22}
3	$s'(X^2+Y^2)$	$-R''_{21}$
4	0	$-R''_{22}$
5	$s(X^2+Y^2)$	$R'_{11}R''_{21} - R''_{11}R'_{21}$
6	0	$R'_{11}R''_{22} - R''_{12}R'_{21}$
7	$s's''X$	$t'_2 - t''_2$
8	$s's''Y$	$-t'_1 + t''_1$
9	$ss''X$	$R'_{11}t''_2 - R'_{21}t''_1$
10	$ss''Y$	$R'_{12}t''_2 - R'_{22}t''_1$
11	$ss'X$	$R''_{21}t'_1 - R''_{11}t'_2$
12	$ss'Y$	$R''_{22}t'_1 - R''_{12}t'_2$
13	$ss's''$	$t'_1t''_2 - t''_1t'_2$

Table 6: Table 1 for data coming from a single central camera.

We observe that the tensor equation has 3 coefficients that are zero: the rank deficiency of the equation system is thus 3 at least (V_2, V_4 and V_6 are unconstrained). Hence, the linear estimation of the tensor will not give a unique solution! We should check if the rank-deficieny is even higher.

We argue that tensor coefficients associated with data coefficients that are different combinations of X, Y, s, s', s'', must be zero. Overall, we thus conclude that the linear equation system for estimating the tensor, has a rank-deficiency of 3, instead of 1 for general data.

Let us look at the solutions of the linear equation system. First, note that the true solution (identity transformations) is represented by the vector:

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Clearly, it is in the null-space of the equation matrix (the non-zero tensor coefficients are the ones that are not constrained by the linear equations).

The null-space of the equation system, when established with data from a central camera, is formed by vectors (for any a, b, c):

$$\begin{pmatrix} 0 \\ a \\ 0 \\ b \\ 0 \\ c \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

A.2 Two Central 2D Cameras

We now extend the previous section by taking into account a second central camera: besides rays as in the previous section, we now consider additional rays passing through a second point (optical center). Let the second point be, without loss of generality:

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Triplets of points on rays going through that point, can be modeled as:

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + \frac{1}{s} \begin{pmatrix} X \\ Y \\ 0 \end{pmatrix} = \begin{pmatrix} X/s \\ 1 + Y/s \\ 1 \end{pmatrix} \sim \begin{pmatrix} X \\ s + Y \\ s \end{pmatrix}$$

Inserting these coordinates in table 1 gives coefficients on the tensor equation that are given in table 7.

i	C_i	V_i
1	$s''\{X^2+(Y+s)(Y+s')\}$	R'_{21}
2	$s''X(s'-s)$	R'_{22}
3	$s'\{X^2+(Y+s)(Y+s'')\}$	$-R''_{21}$
4	$s'X(s''-s)$	$-R''_{22}$
5	$s\{X^2+(Y+s')(Y+s'')\}$	$R'_{11}R''_{21}-R''_{11}R'_{21}$
6	$sX(s''-s')$	$R'_{11}R''_{22}-R''_{12}R'_{21}$
7	$s's''X$	$t'_2-t''_2$
8	$s's''(Y+s)$	$-t'_1+t''_1$
9	$ss''X$	$R'_{11}t''_2-R'_{21}t''_1$
10	$ss''(Y+s')$	$R'_{12}t''_2-R'_{22}t''_1$
11	$ss'X$	$R''_{21}t'_1-R''_{11}t'_2$
12	$ss'(Y+s'')$	$R''_{22}t'_1-R''_{12}t'_2$
13	$ss's''$	$t'_1t''_2-t''_1t'_2$

Table 7: Table 1 for data coming from the second central camera.

We now check which of the null-vectors of the previous section, are also null-vectors of the extended equation system. They have thus to satisfy $C_2a+C_4b+C_6c=0$:

$$X\{s''(s'-s)a+s'(s''-s)b+s(s''-s')c\}=0$$

We exclude $X=0$, and order the other terms:

$$-ss'(b+c)+ss''(c-a)+s's''(a+b)=0$$

This must hold for any values of s, s', s'', thus it must hold:

$$b+c=c-a=a+b=0$$

It follows that $a=c=-b$, which corresponds, up to scale, to the true solution.

In conclusion, a system consisting of two central cameras, allows in general a unique linear solution of the calibration tensor, and can thus be calibrated with the general method.

B Analysis of Underconstrained Cases for 3D Cameras

We now do a similar analysis for 3D cameras. First, we give expressions for all four tensors involved: table 4 is extended and completely given in tables 8 and 9.

One may observe that these tensors share many coefficients. There are exactly 69 different coefficients, that are shared among the four tensors as shown in tables 10 and 11.

C_i	T_i^1	T_i^2	T_i^3	T_i^4
$Q_1Q_1'Q_1''$	0	0	0	$R'_{21}R''_{31} - R'_{31}R''_{21}$
$Q_1Q_1'Q_2''$	0	0	0	$R'_{21}R''_{32} - R'_{31}R''_{22}$
$Q_1Q_1'Q_3''$	0	0	0	$R'_{21}R''_{33} - R'_{31}R''_{23}$
$Q_1Q_1'Q_4''$	0	R'_{31}	R'_{21}	$R'_{21}t''_3 - R'_{31}t''_2$
$Q_1Q_2'Q_1''$	0	0	0	$R'_{22}R''_{31} - R'_{32}R''_{21}$
$Q_1Q_2'Q_2''$	0	0	0	$R'_{22}R''_{32} - R'_{32}R''_{22}$
$Q_1Q_2'Q_3''$	0	0	0	$R'_{22}R''_{33} - R'_{32}R''_{23}$
$Q_1Q_2'Q_4''$	0	R'_{32}	R'_{22}	$R'_{22}t''_3 - R'_{32}t''_2$
$Q_1Q_3'Q_1''$	0	0	0	$R'_{23}R''_{31} - R'_{33}R''_{21}$
$Q_1Q_3'Q_2''$	0	0	0	$R'_{23}R''_{32} - R'_{33}R''_{22}$
$Q_1Q_3'Q_3''$	0	0	0	$R'_{23}R''_{33} - R'_{33}R''_{23}$
$Q_1Q_3'Q_4''$	0	R'_{33}	R'_{23}	$R'_{23}t''_3 - R'_{33}t''_2$
$Q_1Q_4'Q_1''$	0	$-R''_{31}$	$-R''_{21}$	$t'_2R''_{31} - t'_3R''_{21}$
$Q_1Q_4'Q_2''$	0	$-R''_{32}$	$-R''_{22}$	$t'_2R''_{32} - t'_3R''_{22}$
$Q_1Q_4'Q_3''$	0	$-R''_{33}$	$-R''_{23}$	$t'_2R''_{33} - t'_3R''_{23}$
$Q_1Q_4'Q_4''$	0	$t'_3 - t''_3$	$t'_2 - t''_2$	$t'_2t''_3 - t'_3t''_2$
$Q_2Q_1'Q_1''$	0	0	0	$R'_{31}R''_{11} - R'_{11}R''_{31}$
$Q_2Q_1'Q_2''$	0	0	0	$R'_{31}R''_{12} - R'_{11}R''_{32}$
$Q_2Q_1'Q_3''$	0	0	0	$R'_{31}R''_{13} - R'_{11}R''_{33}$
$Q_2Q_1'Q_4''$	R'_{31}	0	$-R'_{11}$	$R'_{31}t''_1 - R'_{11}t''_3$
$Q_2Q_2'Q_1''$	0	0	0	$R'_{32}R''_{11} - R'_{12}R''_{31}$
$Q_2Q_2'Q_2''$	0	0	0	$R'_{32}R''_{12} - R'_{12}R''_{32}$
$Q_2Q_2'Q_3''$	0	0	0	$R'_{32}R''_{13} - R'_{12}R''_{33}$
$Q_2Q_2'Q_4''$	R'_{32}	0	$-R'_{12}$	$R'_{32}t''_1 - R'_{12}t''_3$
$Q_2Q_3'Q_1''$	0	0	0	$R'_{33}R''_{11} - R'_{13}R''_{31}$
$Q_2Q_3'Q_2''$	0	0	0	$R'_{33}R''_{12} - R'_{13}R''_{32}$
$Q_2Q_3'Q_3''$	0	0	0	$R'_{33}R''_{13} - R'_{13}R''_{33}$
$Q_2Q_3'Q_4''$	R'_{33}	0	$-R'_{13}$	$R'_{33}t''_1 - R'_{13}t''_3$
$Q_2Q_4'Q_1''$	$-R''_{31}$	0	R''_{11}	$t'_3R''_{11} - t'_1R''_{31}$
$Q_2Q_4'Q_2''$	$-R''_{32}$	0	R''_{12}	$t'_3R''_{12} - t'_1R''_{32}$
$Q_2Q_4'Q_3''$	$-R''_{33}$	0	R''_{13}	$t'_3R''_{13} - t'_1R''_{33}$
$Q_2Q_4'Q_4''$	$t'_3 - t''_3$	0	$t''_1 - t'_1$	$t'_3t''_1 - t'_1t''_3$

Table 8: Tensor coefficients, part I.

C_i	T_i^1	T_i^2	T_i^3	T_i^4
$Q_3Q_1'Q_1''$	0	0	0	$R_{11}'R_{21}'' - R_{21}'R_{11}''$
$Q_3Q_1'Q_2''$	0	0	0	$R_{11}'R_{22}'' - R_{21}'R_{12}''$
$Q_3Q_1'Q_3''$	0	0	0	$R_{11}'R_{23}'' - R_{21}'R_{13}''$
$Q_3Q_1'Q_4''$	$-R_{21}'$	$-R_{11}'$	0	$R_{11}'t_2'' - R_{21}'t_1''$
$Q_3Q_2'Q_1''$	0	0	0	$R_{12}'R_{21}'' - R_{22}'R_{11}''$
$Q_3Q_2'Q_2''$	0	0	0	$R_{12}'R_{22}'' - R_{22}'R_{12}''$
$Q_3Q_2'Q_3''$	0	0	0	$R_{12}'R_{23}'' - R_{22}'R_{13}''$
$Q_3Q_2'Q_4''$	$-R_{22}'$	$-R_{12}'$	0	$R_{12}'t_2'' - R_{22}'t_1''$
$Q_3Q_3'Q_1''$	0	0	0	$R_{13}'R_{21}'' - R_{23}'R_{11}''$
$Q_3Q_3'Q_2''$	0	0	0	$R_{13}'R_{22}'' - R_{23}'R_{12}''$
$Q_3Q_3'Q_3''$	0	0	0	$R_{13}'R_{23}'' - R_{23}'R_{13}''$
$Q_3Q_3'Q_4''$	$-R_{23}'$	$-R_{13}'$	0	$R_{13}'t_2'' - R_{23}'t_1''$
$Q_3Q_4'Q_1''$	R_{21}''	R_{11}''	0	$t_1'R_{21}'' - t_2'R_{11}''$
$Q_3Q_4'Q_2''$	R_{22}''	R_{12}''	0	$t_1'R_{22}'' - t_2'R_{12}''$
$Q_3Q_4'Q_3''$	R_{23}''	R_{13}''	0	$t_1'R_{23}'' - t_2'R_{13}''$
$Q_3Q_4'Q_4''$	$t_2'' - t_2'$	$t_1'' - t_1'$	0	$t_1't_2'' - t_2't_1''$
$Q_4Q_1'Q_1''$	$R_{21}'R_{31}'' - R_{21}''R_{31}'$	$R_{11}'R_{31}'' - R_{11}''R_{31}'$	$R_{11}'R_{21}'' - R_{21}'R_{11}''$	0
$Q_4Q_1'Q_2''$	$R_{21}'R_{32}'' - R_{22}''R_{31}'$	$R_{11}'R_{32}'' - R_{12}''R_{31}'$	$R_{11}'R_{22}'' - R_{21}'R_{12}''$	0
$Q_4Q_1'Q_3''$	$R_{21}'R_{33}'' - R_{23}''R_{31}'$	$R_{11}'R_{33}'' - R_{13}''R_{31}'$	$R_{11}'R_{23}'' - R_{21}'R_{13}''$	0
$Q_4Q_1'Q_4''$	$R_{21}'t_3'' - R_{31}'t_2''$	$R_{11}'t_3'' - R_{31}'t_1''$	$R_{11}'t_2'' - R_{21}'t_1''$	0
$Q_4Q_2'Q_1''$	$R_{22}'R_{31}'' - R_{21}''R_{32}'$	$R_{12}'R_{31}'' - R_{11}''R_{32}'$	$R_{12}'R_{21}'' - R_{22}'R_{11}''$	0
$Q_4Q_2'Q_2''$	$R_{22}'R_{32}'' - R_{22}''R_{32}'$	$R_{12}'R_{32}'' - R_{12}''R_{32}'$	$R_{12}'R_{22}'' - R_{22}'R_{12}''$	0
$Q_4Q_2'Q_3''$	$R_{22}'R_{33}'' - R_{23}''R_{32}'$	$R_{12}'R_{33}'' - R_{13}''R_{32}'$	$R_{12}'R_{23}'' - R_{22}'R_{13}''$	0
$Q_4Q_2'Q_4''$	$R_{22}'t_3'' - R_{32}'t_2''$	$R_{12}'t_3'' - R_{32}t_1''$	$R_{12}'t_2'' - R_{22}'t_1''$	0
$Q_4Q_3'Q_1''$	$R_{23}'R_{31}'' - R_{21}''R_{33}'$	$R_{13}'R_{31}'' - R_{11}''R_{33}'$	$R_{13}'R_{21}'' - R_{23}'R_{11}''$	0
$Q_4Q_3'Q_2''$	$R_{23}'R_{32}'' - R_{22}''R_{33}'$	$R_{13}'R_{32}'' - R_{12}''R_{33}'$	$R_{13}'R_{22}'' - R_{23}'R_{12}''$	0
$Q_4Q_3'Q_3''$	$R_{23}'R_{33}'' - R_{23}''R_{33}'$	$R_{13}'R_{33}'' - R_{13}''R_{33}'$	$R_{13}'R_{23}'' - R_{23}'R_{13}''$	0
$Q_4Q_3'Q_4''$	$R_{23}'t_3'' - R_{33}'t_2''$	$R_{13}'t_3'' - R_{33}'t_1''$	$R_{13}'t_2'' - R_{23}'t_1''$	0
$Q_4Q_4'Q_1''$	$R_{31}''t_2' - R_{21}''t_3'$	$R_{31}''t_1' - R_{11}''t_3'$	$t_1'R_{21}'' - t_2'R_{11}''$	0
$Q_4Q_4'Q_2''$	$R_{32}''t_2' - R_{22}''t_3'$	$R_{32}''t_1' - R_{12}''t_3'$	$t_1'R_{22}'' - t_2'R_{12}''$	0
$Q_4Q_4'Q_3''$	$R_{33}''t_2' - R_{23}''t_3'$	$R_{33}''t_1' - R_{13}''t_3'$	$t_1'R_{23}'' - t_2'R_{13}''$	0
$Q_4Q_4'Q_4''$	$t_2't_3'' - t_3't_2''$	$t_1't_3'' - t_1''t_3'$	$t_1't_2'' - t_2't_1''$	0

Table 9: Tensor coefficients, part II.

	Coupled motion pars	T^1	T^2	T^3	T^4
1	R'_{11}	0	$-Q_3Q'_1Q''_4$	$-Q_2Q'_1Q''_4$	0
2	R'_{12}	0	$-Q_3Q'_2Q''_4$	$-Q_2Q'_2Q''_4$	0
3	R'_{13}	0	$-Q_3Q'_3Q''_4$	$-Q_2Q'_3Q''_4$	0
4	R'_{21}	$-Q_3Q'_1Q''_4$	0	$Q_1Q'_1Q''_4$	0
5	R'_{22}	$-Q_3Q'_2Q''_4$	0	$Q_1Q'_2Q''_4$	0
6	R'_{23}	$-Q_3Q'_3Q''_4$	0	$Q_1Q'_3Q''_4$	0
7	R'_{31}	$Q_2Q'_1Q''_4$	$Q_1Q'_1Q''_4$	0	0
8	R'_{32}	$Q_2Q'_2Q''_4$	$Q_1Q'_2Q''_4$	0	0
9	R'_{33}	$Q_2Q'_3Q''_4$	$Q_1Q'_3Q''_4$	0	0
10	R''_{11}	0	$Q_3Q'_4Q''_1$	$Q_2Q'_4Q''_1$	0
11	R''_{12}	0	$Q_3Q'_4Q''_2$	$Q_2Q'_4Q''_2$	0
12	R''_{13}	0	$Q_3Q'_4Q''_3$	$Q_2Q'_4Q''_3$	0
13	R''_{21}	$Q_3Q'_4Q''_1$	0	$-Q_1Q'_4Q''_1$	0
14	R''_{22}	$Q_3Q'_4Q''_2$	0	$-Q_1Q'_4Q''_2$	0
15	R''_{23}	$Q_3Q'_4Q''_3$	0	$-Q_1Q'_4Q''_3$	0
16	R''_{31}	$-Q_2Q'_4Q''_1$	$-Q_1Q'_4Q''_1$	0	0
17	R''_{32}	$-Q_2Q'_4Q''_2$	$-Q_1Q'_4Q''_2$	0	0
18	R''_{33}	$-Q_2Q'_4Q''_3$	$-Q_1Q'_4Q''_3$	0	0
19	$t'_1 - t''_1$	0	$-Q_3Q'_4Q''_4$	$-Q_3Q'_4Q''_4$	0
20	$t'_2 - t''_2$	$-Q_3Q'_4Q''_4$	0	$Q_1Q'_4Q''_4$	0
21	$t'_3 - t''_3$	$Q_2Q'_4Q''_4$	$Q_1Q'_4Q''_4$	0	0
22	$R'_{11}R''_{21} - R'_{21}R''_{11}$	0	0	$Q_4Q'_1Q''_1$	$Q_3Q'_1Q''_1$
23	$R'_{11}R''_{22} - R'_{21}R''_{12}$	0	0	$Q_4Q'_1Q''_2$	$Q_3Q'_1Q''_2$
24	$R'_{11}R''_{23} - R'_{21}R''_{13}$	0	0	$Q_4Q'_1Q''_3$	$Q_3Q'_1Q''_3$
25	$R'_{12}R''_{21} - R'_{22}R''_{11}$	0	0	$Q_4Q'_2Q''_1$	$Q_3Q'_2Q''_1$
26	$R'_{12}R''_{22} - R'_{22}R''_{12}$	0	0	$Q_4Q'_2Q''_2$	$Q_3Q'_2Q''_2$
27	$R'_{12}R''_{23} - R'_{22}R''_{13}$	0	0	$Q_4Q'_2Q''_3$	$Q_3Q'_2Q''_3$
28	$R'_{13}R''_{21} - R'_{23}R''_{11}$	0	0	$Q_4Q'_3Q''_1$	$Q_3Q'_3Q''_1$
29	$R'_{13}R''_{22} - R'_{23}R''_{12}$	0	0	$Q_4Q'_3Q''_2$	$Q_3Q'_3Q''_2$
30	$R'_{13}R''_{23} - R'_{23}R''_{13}$	0	0	$Q_4Q'_3Q''_3$	$Q_3Q'_3Q''_3$
31	$R'_{11}R''_{31} - R'_{31}R''_{11}$	0	$Q_4Q'_1Q''_1$	0	$-Q_2Q'_1Q''_1$
32	$R'_{11}R''_{32} - R'_{31}R''_{12}$	0	$Q_4Q'_1Q''_2$	0	$-Q_2Q'_1Q''_2$
33	$R'_{11}R''_{33} - R'_{31}R''_{13}$	0	$Q_4Q'_1Q''_3$	0	$-Q_2Q'_1Q''_3$
34	$R'_{12}R''_{31} - R'_{32}R''_{11}$	0	$Q_4Q'_2Q''_1$	0	$-Q_2Q'_2Q''_1$
35	$R'_{12}R''_{32} - R'_{32}R''_{12}$	0	$Q_4Q'_2Q''_2$	0	$-Q_2Q'_2Q''_2$
36	$R'_{12}R''_{33} - R'_{32}R''_{13}$	0	$Q_4Q'_2Q''_3$	0	$-Q_2Q'_2Q''_3$
37	$R'_{13}R''_{31} - R'_{33}R''_{11}$	0	$Q_4Q'_3Q''_1$	0	$-Q_2Q'_3Q''_1$
38	$R'_{13}R''_{32} - R'_{33}R''_{12}$	0	$Q_4Q'_3Q''_2$	0	$-Q_2Q'_3Q''_2$
39	$R'_{13}R''_{33} - R'_{33}R''_{13}$	0	$Q_4Q'_3Q''_3$	0	$-Q_2Q'_3Q''_3$

Table 10: Coupled tensor coefficients, part I.

	Coupled motion pars	T^1	T^2	T^3	T^4
40	$R'_{21}R''_{31} - R'_{31}R''_{21}$	$Q_4Q'_1Q''_1$	0	0	$Q_1Q'_1Q''_1$
41	$R'_{21}R''_{32} - R'_{31}R''_{22}$	$Q_4Q'_1Q''_2$	0	0	$Q_1Q'_1Q''_2$
42	$R'_{21}R''_{33} - R'_{31}R''_{23}$	$Q_4Q'_1Q''_3$	0	0	$Q_1Q'_1Q''_3$
43	$R'_{22}R''_{31} - R'_{32}R''_{21}$	$Q_4Q'_2Q''_1$	0	0	$Q_1Q'_2Q''_1$
44	$R'_{22}R''_{32} - R'_{32}R''_{22}$	$Q_4Q'_2Q''_2$	0	0	$Q_1Q'_2Q''_2$
45	$R'_{22}R''_{33} - R'_{32}R''_{23}$	$Q_4Q'_2Q''_3$	0	0	$Q_1Q'_2Q''_3$
46	$R'_{23}R''_{31} - R'_{33}R''_{21}$	$Q_4Q'_3Q''_1$	0	0	$Q_1Q'_3Q''_1$
47	$R'_{23}R''_{32} - R'_{33}R''_{22}$	$Q_4Q'_3Q''_2$	0	0	$Q_1Q'_3Q''_2$
48	$R'_{23}R''_{33} - R'_{33}R''_{23}$	$Q_4Q'_3Q''_3$	0	0	$Q_1Q'_3Q''_3$
49	$R'_{11}t''_2 - R'_{21}t''_1$	0	0	$Q_4Q'_1Q''_4$	$Q_3Q'_1Q''_4$
50	$R'_{12}t''_2 - R'_{22}t''_1$	0	0	$Q_4Q'_2Q''_4$	$Q_3Q'_2Q''_4$
51	$R'_{13}t''_2 - R'_{23}t''_1$	0	0	$Q_4Q'_3Q''_4$	$Q_3Q'_3Q''_4$
52	$R'_{11}t''_3 - R'_{31}t''_1$	0	$Q_4Q'_1Q''_4$	0	$-Q_2Q'_1Q''_4$
53	$R'_{12}t''_3 - R'_{32}t''_1$	0	$Q_4Q'_2Q''_4$	0	$-Q_2Q'_2Q''_4$
54	$R'_{13}t''_3 - R'_{33}t''_1$	0	$Q_4Q'_3Q''_4$	0	$-Q_2Q'_3Q''_4$
55	$R'_{21}t''_3 - R'_{31}t''_2$	$Q_4Q'_1Q''_4$	0	0	$-Q_1Q'_1Q''_4$
56	$R'_{22}t''_3 - R'_{32}t''_2$	$Q_4Q'_2Q''_4$	0	0	$-Q_1Q'_2Q''_4$
57	$R'_{23}t''_3 - R'_{33}t''_2$	$Q_4Q'_3Q''_4$	0	0	$-Q_1Q'_3Q''_4$
58	$R''_{11}t'_2 - R''_{21}t'_1$	0	0	$-Q_4Q'_4Q''_1$	$-Q_3Q'_4Q''_1$
59	$R''_{12}t'_2 - R''_{22}t'_1$	0	0	$-Q_4Q'_4Q''_2$	$-Q_3Q'_4Q''_2$
60	$R''_{13}t'_2 - R''_{23}t'_1$	0	0	$-Q_4Q'_4Q''_3$	$-Q_3Q'_4Q''_3$
61	$R''_{11}t'_3 - R''_{31}t'_1$	0	$-Q_4Q'_4Q''_1$	0	$Q_2Q'_4Q''_1$
62	$R''_{12}t'_3 - R''_{32}t'_1$	0	$-Q_4Q'_4Q''_2$	0	$Q_2Q'_4Q''_2$
63	$R''_{13}t'_3 - R''_{33}t'_1$	0	$-Q_4Q'_4Q''_3$	0	$Q_2Q'_4Q''_3$
64	$R''_{21}t'_3 - R''_{31}t'_2$	$-Q_4Q'_4Q''_1$	0	0	$Q_1Q'_4Q''_1$
65	$R''_{22}t'_3 - R''_{32}t'_2$	$-Q_4Q'_4Q''_2$	0	0	$Q_1Q'_4Q''_2$
66	$R''_{23}t'_3 - R''_{33}t'_2$	$-Q_4Q'_4Q''_3$	0	0	$Q_1Q'_4Q''_3$
67	$t'_1t''_2 - t'_2t''_1$	0	0	$Q_4Q'_4Q''_4$	$Q_3Q'_4Q''_4$
68	$t'_1t''_3 - t'_3t''_1$	0	$Q_4Q'_4Q''_4$	0	$-Q_2Q'_4Q''_4$
69	$t'_2t''_3 - t'_3t''_2$	$Q_4Q'_4Q''_4$	0	0	$Q_1Q'_4Q''_4$

Table 11: Coupled tensor coefficients, part II.

B.1 A Single Central 3D Camera

Aligned points are parameterized as:

$$\mathbf{Q} \sim \begin{pmatrix} X \\ Y \\ Z \\ s \end{pmatrix} \qquad \mathbf{Q}' \sim \begin{pmatrix} X \\ Y \\ Z \\ s' \end{pmatrix} \qquad \mathbf{Q}'' \sim \begin{pmatrix} X \\ Y \\ Z \\ s'' \end{pmatrix}$$

Inserting this in tables 10 and 11 gives tables 12 and 13. The last column of these tables indicates if tensor coefficients are estimated as being zero, and which of the four tensors allows this to be done. The underlying reasoning is as follows: consider coefficient 22, and the entry associated with the 3rd tensor: sX^2. This term appears in no other coefficient associated with that tensor, which is why coefficient 22 must be zero.

There are a total of 24 undetermined coefficients: 1, 5, 9, 10, 14, 18, 23, 24, 25, 27, 28, 29, 32, 33, 34, 36, 37, 38, 41, 42, 43, 45, 46, 47.

For the four tensors, we now list available constraints on these:

1 (5-9=0) (14-18=0) (41+43=0) (42+46=0) (45+47=0)

2 (1-9=0) (10-18=0) (32+34=0) (33+37=0) (36+38=0)

3 (1-5=0) (10-14=0) (23+25=0) (24+28=0) (27+29=0)

4 (42+46=0) (23+25-33-37+45+47=0) (24+28=0) (36+38=0) (27+29=0) (41+43=0) (32+34=0)

Note that some equations are redundant.

	Coupled motion pars	T^1	T^2	T^3	T^4	ZERO DUE TO
1	R'_{11}	0	$-s''XZ$	$-s''XY$	0	
2	R'_{12}	0	$-s''YZ$	$-s''Y^2$	0	2, 3
3	R'_{13}	0	$-s''Z^2$	$-s''YZ$	0	2, 3
4	R'_{21}	$-s''XZ$	0	$s''X^2$	0	1, 3
5	R'_{22}	$-s''YZ$	0	$s''XY$	0	
6	R'_{23}	$-s''Z^2$	0	$s''XZ$	0	1, 3
7	R'_{31}	$s''XY$	$s''X^2$	0	0	1, 2
8	R'_{32}	$s''Y^2$	$s''XY$	0	0	1, 2
9	R'_{33}	$s''YZ$	$s''XZ$	0	0	
10	R''_{11}	0	$s'XZ$	$s'XY$	0	
11	R''_{12}	0	$s'YZ$	$s'Y^2$	0	2, 3
12	R''_{13}	0	$s'Z^2$	$s'YZ$	0	2, 3
13	R''_{21}	$s'XZ$	0	$-s'X^2$	0	1, 3
14	R''_{22}	$s'YZ$	0	$-s'XY$	0	
15	R''_{23}	$s'Z^2$	0	$-s'XZ$	0	1, 3
16	R''_{31}	$-s'XY$	$-s'X^2$	0	0	1, 2
17	R''_{32}	$-s'Y^2$	$-s'XY$	0	0	1, 2
18	R''_{33}	$-s'YZ$	$-s'XZ$	0	0	
19	$t'_1 - t''_1$	0	$-s's''Z$	$-s's''Z$	0	2, 3
20	$t'_2 - t''_2$	$-s's''Z$	0	$s's''X$	0	1, 3
21	$t'_3 - t''_3$	$s's''Y$	$s's''X$	0	0	1, 2
22	$R'_{11}R''_{21} - R'_{21}R''_{11}$	0	0	sX^2	X^2Z	3
23	$R'_{11}R''_{22} - R'_{21}R''_{12}$	0	0	sXY	XYZ	
24	$R'_{11}R''_{23} - R'_{21}R''_{13}$	0	0	sXZ	XZ^2	
25	$R'_{12}R''_{21} - R'_{22}R''_{11}$	0	0	sXY	XYZ	
26	$R'_{12}R''_{22} - R'_{22}R''_{12}$	0	0	sY^2	Y^2Z	3
27	$R'_{12}R''_{23} - R'_{22}R''_{13}$	0	0	sYZ	YZ^2	
28	$R'_{13}R''_{21} - R'_{23}R''_{11}$	0	0	sXZ	XZ^2	
29	$R'_{13}R''_{22} - R'_{23}R''_{12}$	0	0	sYZ	YZ^2	
30	$R'_{13}R''_{23} - R'_{23}R''_{13}$	0	0	sZ^2	Z^3	3, 4
31	$R'_{11}R''_{31} - R'_{31}R''_{11}$	0	sX^2	0	$-X^2Y$	2
32	$R'_{11}R''_{32} - R'_{31}R''_{12}$	0	sXY	0	$-XY^2$	
33	$R'_{11}R''_{33} - R'_{31}R''_{13}$	0	sXZ	0	$-XYZ$	
34	$R'_{12}R''_{31} - R'_{32}R''_{11}$	0	sXY	0	$-XY^2$	
35	$R'_{12}R''_{32} - R'_{32}R''_{12}$	0	sY^2	0	$-Y^3$	2, 4
36	$R'_{12}R''_{33} - R'_{32}R''_{13}$	0	sYZ	0	$-Y^2Z$	
37	$R'_{13}R''_{31} - R'_{33}R''_{11}$	0	sXZ	0	$-XYZ$	
38	$R'_{13}R''_{32} - R'_{33}R''_{12}$	0	sYZ	0	$-Y^2Z$	
39	$R'_{13}R''_{33} - R'_{33}R''_{13}$	0	sZ^2	0	$-YZ^2$	2

Table 12: Coupled tensor coefficients, for data coming from a single central camera, part I.

	Coupled motion pars	T^1	T^2	T^3	T^4	ZERO DUE TO
40	$R'_{21}R''_{31} - R'_{31}R''_{21}$	sX^2	0	0	X^3	1, 4
41	$R'_{21}R''_{32} - R'_{31}R''_{22}$	sXY	0	0	X^2Y	
42	$R'_{21}R''_{33} - R'_{31}R''_{23}$	sXZ	0	0	X^2Z	
43	$R'_{22}R''_{31} - R'_{32}R''_{21}$	sXY	0	0	X^2Y	
44	$R'_{22}R''_{32} - R'_{32}R''_{22}$	sY^2	0	0	XY^2	1
45	$R'_{22}R''_{33} - R'_{32}R''_{23}$	sYZ	0	0	XYZ	
46	$R'_{23}R''_{31} - R'_{33}R''_{21}$	sXZ	0	0	X^2Z	
47	$R'_{23}R''_{32} - R'_{33}R''_{22}$	sYZ	0	0	XYZ	
48	$R'_{23}R''_{33} - R'_{33}R''_{23}$	sZ^2	0	0	XZ^2	1
49	$R'_{11}t''_2 - R'_{21}t''_1$	0	0	$ss''X$	$s''XZ$	3
50	$R'_{12}t''_2 - R'_{22}t''_1$	0	0	$ss''Y$	$s''YZ$	3
51	$R'_{13}t''_2 - R'_{23}t''_1$	0	0	$ss''Z$	$s''Z^2$	3, 4
52	$R'_{11}t''_3 - R'_{31}t''_1$	0	$ss''X$	0	$-s''XY$	2
53	$R'_{12}t''_3 - R'_{32}t''_1$	0	$ss''Y$	0	$-s''Y^2$	2, 4
54	$R'_{13}t''_3 - R'_{33}t''_1$	0	$ss''Z$	0	$-s''YZ$	2
55	$R'_{21}t''_3 - R'_{31}t''_2$	$ss''X$	0	0	$-s''X^2$	1, 4
56	$R'_{22}t''_3 - R'_{32}t''_2$	$ss''Y$	0	0	$-s''XY$	1
57	$R'_{23}t''_3 - R'_{33}t''_2$	$ss''Z$	0	0	$-s''XZ$	1
58	$R''_{11}t'_2 - R''_{21}t'_1$	0	0	$-ss'X$	$-s'XZ$	3
59	$R''_{12}t'_2 - R''_{22}t'_1$	0	0	$-ss'Y$	$-s'YZ$	3
60	$R''_{13}t'_2 - R''_{23}t'_1$	0	0	$-ss'Z$	$-s'Z^2$	3, 4
61	$R''_{11}t'_3 - R''_{31}t'_1$	0	$-ss'X$	0	$s'XY$	2
62	$R''_{12}t'_3 - R''_{32}t'_1$	0	$-ss'Y$	0	$s'Y^2$	2, 4
63	$R''_{13}t'_3 - R''_{33}t'_1$	0	$-ss'Z$	0	$s'YZ$	2
64	$R''_{21}t'_3 - R''_{31}t'_2$	$-ss'X$	0	0	$s'X^2$	1, 4
65	$R''_{22}t'_3 - R''_{32}t'_2$	$-ss'Y$	0	0	$s'XY$	1
66	$R''_{23}t'_3 - R''_{33}t'_2$	$-ss'Z$	0	0	$s'XZ$	1
67	$t'_1t''_2 - t'_2t''_1$	0	0	$ss's''$	$s's''Z$	3, 4
68	$t'_1t''_3 - t'_3t''_1$	0	$ss's''$	0	$-s's''Y$	2, 4
69	$t'_2t''_3 - t'_3t''_2$	$ss's''$	0	0	$s's''X$	1, 4

Table 13: Coupled tensor coefficients, for data coming from a single central camera, part II.

We may regroup the equations and split them in connected parts:

$$\begin{pmatrix} & 1 & -1 \\ 1 & & -1 \\ 1 & -1 & \end{pmatrix} \begin{pmatrix} 1 \\ 5 \\ 9 \end{pmatrix} = \mathbf{0}$$

$$\begin{pmatrix} & 1 & -1 \\ 1 & & -1 \\ 1 & -1 & \end{pmatrix} \begin{pmatrix} 10 \\ 14 \\ 18 \end{pmatrix} = \mathbf{0}$$

$$\begin{pmatrix} 1 & & 1 & & \\ & 1 & & 1 & \\ & & 1 & & 1 \end{pmatrix} \begin{pmatrix} 23 \\ 24 \\ 25 \\ 27 \\ 28 \\ 29 \end{pmatrix} = \mathbf{0}$$

$$\begin{pmatrix} 1 & & 1 & & \\ & 1 & & 1 & \\ & & 1 & & 1 \end{pmatrix} \begin{pmatrix} 32 \\ 33 \\ 34 \\ 36 \\ 37 \\ 38 \end{pmatrix} = \mathbf{0}$$

$$\begin{pmatrix} 1 & & 1 & & \\ & 1 & & 1 & \\ & & 1 & & 1 \end{pmatrix} \begin{pmatrix} 41 \\ 42 \\ 43 \\ 45 \\ 46 \\ 47 \end{pmatrix} = \mathbf{0}$$

We conclude the following: the unknowns are determined up to $24 - 2 \times 2 - 3 \times 3 - 1 = 10$ degrees of freedom (1 for freedom of scale). We now consider the use of additional data, corresponding to rays passing through a second optical center.

B.2 Two Central 3D Cameras

For simplicity, we only consider tensor coefficients that were undetermined in the previous section.

We again consider already aligned points. Besides rays as in the previous section, we now consider additional rays passing through a second point. Let the second point be, without loss of generality:

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Triplets of points on rays going through that point, can be modeled as:

$$\begin{pmatrix}0\\0\\1\\1\end{pmatrix} + \frac{1}{s}\begin{pmatrix}X\\Y\\Z\\0\end{pmatrix} = \begin{pmatrix}X/s\\Y/s\\1+Z/s\\1\end{pmatrix} \sim \begin{pmatrix}X\\Y\\s+Z\\s\end{pmatrix}$$

We insert this in tables 12 and 13 (only for the coefficients that remained undetermined), which gives table 14.

	T^1	T^2	T^3	T^4
1	0	$-ss''X - s''XZ$	$-s''XY$	0
5	$-ss''Y - s''YZ$	0	$s''XY$	0
9	$s's''Y + s''YZ$	$s's''X + s''XZ$	0	0
10	0	$ss'X + s'XZ$	$s'XY$	0
14	$ss'Y + s'YZ$	0	$-s'XY$	0
18	$-s's''Y - s'YZ$	$-s's''X - s'XZ$	0	0
23	0	0	sXY	$sXY + XYZ$
24	0	0	$ss''X + sXZ$	$ss''X + sXZ + s''XZ + XZ^2$
25	0	0	sXY	$sXY + XYZ$
27	0	0	$ss''Y + sYZ$	$ss''Y + sYZ + s''YZ + YZ^2$
28	0	0	$ss'X + sXZ$	$ss'X + sXZ + s'XZ + XZ^2$
29	0	0	$ss'Y + sYZ$	$ss'Y + sYZ + s'YZ + YZ^2$
32	0	sXY	0	$-XY^2$
33	0	$ss''X + sXZ$	0	$-s''XY - XYZ$
34	0	sXY	0	$-XY^2$
36	0	$ss''Y + sYZ$	0	$-s''Y^2 - Y^2Z$
37	0	$ss'X + sXZ$	0	$-s'XY - XYZ$
38	0	$ss'Y + sYZ$	0	$-s'Y^2 - Y^2Z$
41	sXY	0	0	X^2Y
42	$ss''X + sXZ$	0	0	$s''X^2 + X^2Z$
43	sXY	0	0	X^2Y
45	$ss''Y + sYZ$	0	0	$s''XY + XYZ$
46	$ss'X + sXZ$	0	0	$s'X^2 + X^2Z$
47	$ss'Y + sYZ$	0	0	$s'XY + XYZ$

Table 14: Coupled tensor coefficients, for data coming from a second central camera.

Let us group terms with equal monomials in (X, Y, Z, s, s', s''), per tensor:

T^1	
$ss'X$	46
$ss'Y$	14+47
$ss''X$	42
$ss''Y$	45-5
sXY	41+43
sXZ	42+46
sYZ	45+47
$s's''Y$	9-18
$s'YZ$	14-18
$s''YZ$	9-5

T^2	
$ss'X$	10+37
$ss'Y$	38
$ss''X$	33-1
$ss''Y$	36
sXY	32+34
sXZ	33+37
sYZ	36+38
$s's''X$	9-18
$s'XZ$	10-18
$s''XZ$	9-1

T^3	
$ss'X$	28
$ss'Y$	29
$ss''X$	24
$ss''Y$	27
sXY	23+25
sXZ	24+28
sYZ	27+29
$s'XY$	10-14
$s''XY$	5-1

T^4	
$ss'X$	28
$ss'Y$	29
$ss''X$	24
$ss''Y$	27
sXY	23+25
sXZ	24+28
sYZ	27+29
$s'X^2$	46
$s'XY$	47-37
$s'XZ$	28
$s'Y^2$	-38
$s'YZ$	29
$s''X^2$	42
$s''XY$	45-33
$s''XZ$	24
$s''Y^2$	-36
$s''YZ$	27
X^2Y	41+43
X^2Z	42+46
XY^2	-32-34
XYZ	23+25-33-37+45+47
XZ^2	24+28
Y^2Z	-36-38
YZ^2	27+29

Figure 12: Grouped coefficients of table 14

.

All coefficients that appear alone, must be zero: 24, 27, 28, 29, 36, 38, 42, 46

We remain with the following 16 coefficients: 1, 5, 9, 10, 14, 18, 23, 25, 32, 33, 34, 37, 41, 43, 45, 47.

We retain the equations given at the end of the previous section, and add the ones given in the tables above:

$$\left(\begin{array}{cccccccccccccccc}
 & 1 & -1 & & & & & & & & & & & & & \\
1 & & -1 & & & & & & & & & & & & & \\
1 & -1 & & & & & & & & & & & & & & \\
 & & & & 1 & -1 & & & & & & & & & & \\
 & & & 1 & & -1 & & & & & & & & & & \\
 & & & 1 & -1 & & & & & & & & & & & \\
 & & & & & & 1 & 1 & & & & & & & & \\
 & & & & & & & & 1 & & 1 & & & & & \\
 & & & & & & & & & 1 & & 1 & & & & \\
 & & & & & & & & & & & & 1 & 1 & & \\
\hline
 & & & & 1 & & & & & & & & & & 1 & \\
 & -1 & & & & & & & & & & & 1 & & & \\
 & & 1 & & & -1 & & & & & & & & & & \\
 & & & 1 & & & & & & & 1 & & & & & \\
-1 & & & & & & & & 1 & & & & & & & \\
 & & & & & & & & & -1 & & & & 1 & \\
 & & & & & & & & -1 & & & & 1 & & &
\end{array}\right)
\left(\begin{array}{c} 1 \\ 5 \\ 9 \\ 10 \\ 14 \\ 18 \\ 23 \\ 25 \\ 32 \\ 33 \\ 34 \\ 37 \\ 41 \\ 43 \\ 45 \\ 47 \end{array}\right) = \mathbf{0}$$

This is seen to be equivalent to:

$$\begin{aligned} 1 = 5 = 9 = 10 = 14 = 18 = 33 = -37 = -45 = -47 \\ 23 = -25 \\ 32 = 34 \\ 41 = 43 \end{aligned}$$

We thus still have a 3-dof ambiguity (4 minus 1 for scale) on the remaining 16 unknown coefficients.

B.3 Three Central 3D Cameras

By repeating the same analysis with an additional, third central camera (not shown here), we find that the ambiguity disappears. In conclusion, a stereo system consisting of 3 or more central cameras, can be calibrated with the general algorithm.

References

[1] E.H. Adelson, J.R. Bergen. The Plenoptic Function and the Elements of Early Vision. In *Computational Models of Visual Processing*, M. Landy and J.A. Movshon (editors), pp. 3-20, MIT Press, 1991.

[2] S. Baker, S. Nayar. A Theory of Catadioptric Image Formation. In *ICCV*, 1998.

[3] H. Bakstein, T. Pajdla. An overview of non-central cameras. In *In Proceedings of Computer Vision Winter Workshop*, Ljubljana, Slovenia, 2001.

[4] P. Brand. Reconstruction tridimensionnelle d'une scène à partir d'une caméra en mouvement : de l'influence de la précision. PhD Thesis, Université Claude Bernard, Lyon, October 1995.

[5] C. Geyer, K. Daniilidis. Paracatadioptric Camera Calibration. In *PAMI*, 2002.

[6] S.J. Gortler, R. Grzeszczuk, R. Szeliski, M.F. Cohen. The Lumigraph. In *SIGGRAPH*, 1996.

[7] M.D. Grossberg, S.K. Nayar. A general imaging model and a method for finding its parameters. In *ICCV*, 2001.

[8] R.I. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[9] R.A. Hicks, R. Bajcsy. Catadioptric Sensors that Approximate Wide-angle Perspective Projections. In *CVPR*, pp. 545-551, 2000.

[10] M. Levoy, P. Hanrahan. Light field rendering. In *SIGGRAPH*, 1996.

[11] J. Neumann, C. Fermüller, Y. Aloimonos. Polydioptric Camera Design and 3D Motion Estimation. In *CVPR*, 2003.

[12] T. Pajdla. Stereo with oblique cameras. *IJCV*, 47(1), 2002.

[13] S. Peleg, M. Ben-Ezra, Y. Pritch. OmniStereo: Panoramic Stereo Imaging. In *PAMI*, pp. 279-290, March 2001.

[14] R. Pless. Using Many Cameras as One. In *CVPR*, 2003.

[15] S. Seitz. The space of all stereo images. In *ICCV*, 2001.

[16] C.C. Slama (editor). *Manual of Photogrammetry*. Fourth Edition, ASPRS, 1980.

[17] A. Shashua, L. Wolf. Homography Tensors: On Algebraic Entities That Represent Three Views of Static or Moving Planar Points. In *ECCV*, 2000.

[18] H.-Y. Shum, A. Kalai, S.M. Seitz. Omnivergent Stereo. In *ICCV*, 1999.

[19] P. Sturm, S. Maybank. On Plane-Based Camera Calibration. In *CVPR*, 1999.

[20] R. Swaminathan, M.D. Grossberg, S.K. Nayar. A perspective on distortions. In *CVPR*, 2003.

[21] Y. Wexler, A.W. Fitzgibbon, A. Zisserman. Learning epipolar geometry from image sequences. In *CVPR*, 2003.

[22] D. Wood, A. Finkelstein, J.F. Hughes, C.E. Thayer, D.H. Salesin. Multiperspective panoramas for cell animation. *SIGGRAPH*, 1997.

[23] Z. Zhang. A flexible new technique for camera calibration. In *PAMI*, 22(11), 2000.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399