On Calibration, Structure from Motion and Multi-View Geometry for Generic Camera Models

Peter Sturm, Srikumar Ramalingam, Suresh K. Lodha

Peter Sturm[1], Srikumar Ramalingam[2], and Suresh Lodha[2]

[1]*INRIA Rhône-Alpes, 655 Avenue de l'Europe, 38330 Montbonnot, France*
Peter.Sturm@inrialpes.fr

[2]*Dept. of Computer Science, University of California, Santa Cruz, USA*
{srikumar,lodha}@cse.ucsc.edu

Abstract We consider calibration and structure from motion tasks for a previously introduced, highly general imaging model, where cameras are modeled as possibly unconstrained sets of projection rays. This allows to describe most existing camera types (at least for those operating in the visible domain), including pinhole cameras, sensors with radial or more general distortions, catadioptric cameras (central or non-central), etc. Generic algorithms for calibration and structure from motion tasks (pose and motion estimation and 3D point triangulation) are outlined. The foundation for a multi-view geometry of non-central cameras is given, leading to the formulation of multi-view matching tensors, analogous to the fundamental matrices, trifocal and quadrifocal tensors of perspective cameras. Besides this, we also introduce a natural hierarchy of camera models: the most general model has unconstrained projection rays whereas the most constrained model dealt with here is the central model, where all rays pass through a single point.

Keywords: Calibration, motion estimation, 3D reconstruction, camera models, non-central cameras.

ON CALIBRATION, STRUCTURE FROM MOTION AND MULTI-VIEW GEOMETRY FOR GENERIC CAMERA MODELS

Peter Sturm[1], Srikumar Ramalingam[2], and Suresh Lodha[2]
[1] *INRIA Rhône-Alpes,* [2] *University of California, Santa Cruz*

Abstract We consider calibration and structure from motion tasks for a previously introduced, highly general imaging model, where cameras are modeled as possibly unconstrained sets of projection rays. This allows to describe most existing camera types (at least for those operating in the visible domain), including pinhole cameras, sensors with radial or more general distortions, catadioptric cameras (central or non-central), etc. Generic algorithms for calibration and structure from motion tasks (pose and motion estimation and 3D point triangulation) are outlined. The foundation for a multi-view geometry of non-central cameras is given, leading to the formulation of multi-view matching tensors, analogous to the fundamental matrices, trifocal and quadrifocal tensors of perspective cameras. Besides this, we also introduce a natural hierarchy of camera models: the most general model has unconstrained projection rays whereas the most constrained model dealt with here is the central model, where all rays pass through a single point.

Keywords: Calibration, motion estimation, 3D reconstruction, camera models, non-central cameras.

1. Introduction

Many different types of cameras including pinhole, stereo, catadioptric, omnidirectional and non-central cameras have been used in computer vision. 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 or structure from motion procedures are often taylor-made for specific camera models, see examples e.g. in [4, 15, 9].

The aim of this work is to relax these constraints: we want to propose and develop calibration and structure from motion methods that should work for any type of camera model, and especially also for cameras without a single effective viewpoint. To do so, we first renounce on parametric models, and

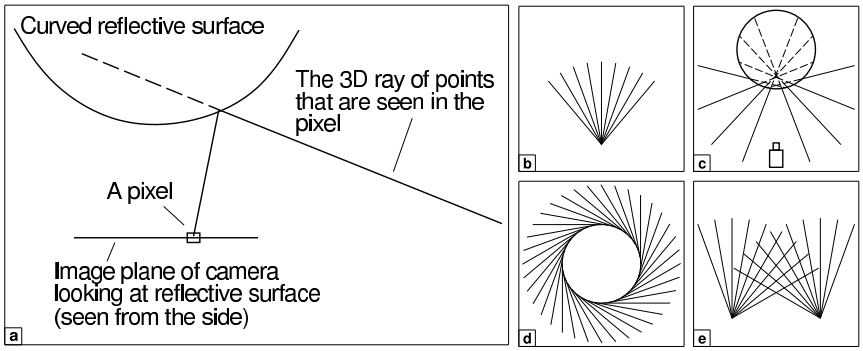


Figure 1. Examples of imaging systems. (a) Catadioptric system. Note that camera rays do not pass through their associated pixels. (b) Central camera (e.g. perspective, with or without radial distortion). (c) Camera looking at reflective sphere. This is a non-central device (camera rays are not intersecting in a single point). (d) Omnivergent imaging system [24, 27]. (e) Stereo system (non-central) consisting of two central cameras.

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 [11]:

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

This general imaging model allows to describe virtually any camera that captures light rays travelling along straight lines. Examples are (cf. figure 1):

- 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 [16]. Such systems, as opposed to central catadioptric systems [1, 8] composed of cameras and parabolic mirrors, do not in general have a single effective 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, many of them being non-central, see e.g. [2, 3, 19, 23, 24, 27, 31, 32], insect eyes, etc.

In this article, we first review some recent work on calibration and structure from motion for this general camera model. Concretely, we outline basics for calibration, pose and motion estimation, as well as 3D point triangulation. We then describe the foundations for a mult-view geometry of the general, non-central camera model, leading to the formulation of multi-view matching tensors, analogous to the fundamental matrices, trifocal and quadrifocal tensors of perspective cameras. Besides this, we also introduce a natural hierarchy of camera models: the most general model has unconstrained projection rays whereas the most constrained model dealt with here is the central model, where all rays pass through a single point. An intermediate model is what we term *axial cameras*: cameras for which there exists a 3D line that cuts all projection rays. This encompasses for example x-slit projections, linear pushbroom cameras and some non-central catadioptric systems. Hints will be given how to adopt the multi-view geometry proposed for the general imaging model, to such axial cameras.

The paper is organized as follows. Section 2 explains some background on Plücker coordinates for 3D lines, which are used to parameterize camera rays in this work. A hierarchy of camera models is proposed in section 3. Sections 4 to 7 deal with calibration, pose estimation, motion estimation, as well as 3D point triangulation. The multi-view geometry for the general camera model is given in section 8. A few experimental results on calibration, motion estimation and 3D reconstruction are shown in section 9.

2. Plücker Coordinates

We represent projection rays as 3D lines, via Plücker coordinates. There exist different definitions for them, the one we use is explained in the following.

Let $\mathbf{A}$ and $\mathbf{B}$ be two 3D points given by homogeneous coordinates, defining a line in 3D. The line can be represented by the skew-symmetric 4×4 Plücker matrix

$$
\begin{aligned}
\mathsf{L} &= \mathbf{A}\mathbf{B}^{\mathsf{T}} - \mathbf{B}\mathbf{A}^{\mathsf{T}} \\
&= \begin{pmatrix}
0 & A_1B_2 - A_2B_1 & A_1B_3 - A_3B_1 & A_1B_4 - A_4B_1 \\
A_2B_1 - A_1B_2 & 0 & A_2B_3 - A_3B_2 & A_2B_4 - A_4B_2 \\
A_3B_1 - A_1B_3 & A_3B_2 - A_2B_3 & 0 & A_3B_4 - A_4B_3 \\
A_4B_1 - A_1B_4 & A_4B_2 - A_2B_4 & A_4B_3 - A_3B_4 & 0
\end{pmatrix}
\end{aligned}
$$

Note that the Plücker matrix is independent (up to scale) of which pair of points on the line are chosen to represent it.

An alternative representation for the line is by its Plücker coordinate vector of length 6:

$$\mathbf{L} = \begin{pmatrix} A_4B_1 - A_1B_4 \\ A_4B_2 - A_2B_4 \\ A_4B_3 - A_3B_4 \\ A_3B_2 - A_2B_3 \\ A_1B_3 - A_3B_1 \\ A_2B_1 - A_1B_2 \end{pmatrix} \tag{1}$$

The Plücker coordinate vector can be split in two 3-vectors $\mathbf{a}$ and $\mathbf{b}$ as follows:

$$\mathbf{a} = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} L_4 \\ L_5 \\ L_6 \end{pmatrix}$$

They satisfy the so-called Plücker constraint: $\mathbf{a}^\mathsf{T}\mathbf{b} = 0$. Furthermore, the Plücker matrix can now be conveniently written as

$$\mathsf{L} = \begin{pmatrix} [\mathbf{b}]_\times & -\mathbf{a} \\ \mathbf{a}^\mathsf{T} & 0 \end{pmatrix}$$

where $[\mathbf{b}]_\times$ is the 3×3 skew-symmetric matrix associated with the cross-product and defined by: $\mathbf{b} \times \mathbf{y} = [\mathbf{b}]_\times \mathbf{y}$.

Consider a metric transformation defined by a rotation matrix R and a translation vector $\mathbf{t}$, acting on points via:

$$\mathbf{C} \to \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \mathbf{C}$$

Plücker coordinates are then transformed according to

$$\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \to \begin{pmatrix} \mathsf{R} & 0 \\ -[\mathbf{t}]_\times \mathsf{R} & \mathsf{R} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}$$

3. A Natural Hierarchy of Camera Models

A **non-central camera** may have completely unconstrained projection rays, whereas for a **central camera**, there exists a point – the **optical center** – that lies on all projection rays. An intermediate case is what we call **axial cameras**, where there exists a line that cuts all projection rays – the **camera axis** (not to be confounded with optical axis). Examples of cameras falling into this class are pushbroom cameras (if motion is translational) [13], x-slit cameras [22, 33], and non-central catadioptric cameras of the following construction: the mirror is any surface of revolution and the optical center of the central camera (can be any central camera, i.e. not necessarily a pinhole) looking at

the mirror lies on its axis of revolution. It is easy to verify that in this case, all projection rays cut the mirror's axis of revolution, i.e. the camera is an axial camera, with the mirror's axis of revolution as camera axis.

These three classes of camera models may also be defined as: existence of a linear space of d dimensions that has an intersection with all projection rays. In this sense, $d = 0$ defines central cameras, $d = 1$ axial cameras and $d = 2$ general non-central cameras.

Intermediate classes do exist. X-slit cameras are a special case of axial cameras: there actually exist 2 lines in space that both cut all projection rays. Similarly, central 1D cameras (cameras with a single row of pixels) can be defined by a point and a line in 3D. Camera models, some of which do not have much practical importance, are summarized in table 1.

Points/lines cutting the rays	Description
None	Non-central camera
1 point	Central camera
2 points	Camera with a single projection ray
1 line	Axial camera
1 point, 1 line	Central 1D camera
2 skew lines	X-slit camera
2 coplanar lines	Union of a non-central 1D camera and a central camera
3 coplanar lines without a common point	Non-central 1D camera

Table 1. Camera models, defined by 3D points and lines that have an intersection with all projection rays of a camera.

It is worthwhile to consider different classes due to the following observation: the usual calibration and motion estimation algorithms proceed by first estimating a matrix or tensor by solving linear equation systems (e.g. the calibration tensors in [30] or the essential matrix [25]). Then, the parameters that are searched for (usually, motion parameters), are extracted from these. However, when estimating for example the 6×6 essential matrix of *non-central* cameras based on image correspondences obtained from *central* or *axial* cameras, then the associated linear equation system does not give a unique solution. Consequently, the algorithms for extracting the actual motion parameters, can not be applied without modification. This is the reason why in [29, 30] we already introduced generic calibration algorithms for both, central and non-central cameras.

In the following, we only deal with central, axial and non-central cameras. Structure from motion computations and multi-view geometry, will be formulated in terms of the Plücker coordinates of camera rays. As for *central cameras*, all rays go through a single point, the optical center. Choosing a local coordinate system with the optical center at the origin, leads to projection rays

whose Plücker sub-vector $\mathbf{b}$ is zero, i.e. the projection rays are of the form:

$$\mathbf{L} = \begin{pmatrix} \mathbf{a} \\ \mathbf{0} \end{pmatrix}$$

This is one reason why the multi-linear matching tensors, e.g. the fundamental matrix, have a "base size" of 3.

As for *axial cameras*, all rays touch a line, the camera axis. Again, by choosing local coordinate systems appropriately, the formulation of the multi-view relations may be simplified, as shown in the following. Assume that the camera axis is the Z-axis. Then, all projection rays have Plücker coordinates with $L_6 = b_3 = 0$:

$$\mathbf{L} = \begin{pmatrix} \mathbf{a} \\ b_1 \\ b_2 \\ 0 \end{pmatrix}$$

Multi-view relations can thus be formulated via tensors of "base size" 5, i.e. the essential matrix for axial cameras will be of size 5×5 (see in later sections).

As for *general non-central cameras*, no such simplification occurs, and multi-view tensors will have "base size" 6.

4. Calibration

We briefly review a generic calibration approach developed in [30], an extension of [5, 10, 11], to calibrate different camera systems. As mentioned, calibration consists in determining, for every pixel, the 3D projection ray associated with it. In [11], this is done as follows: two images of a calibration object with known structure are taken. We suppose that for every pixel, we can determine the point on the calibration object, that is seen by that pixel. For each pixel in the image, we thus obtain two 3D points. Their coordinates are usually only known in a coordinate frame attached to the calibration object; however, if one knows the motion between the two object positions, one can align the coordinate frames. Then, every pixel's projection ray can be computed by simply joining the two observed 3D points.

In [30], we propose a more general approach, that does not require knowledge of the calibration object's displacement. In that case, three images need to be taken at least. The fact that all 3D points observed by a pixel in different views, are on a line in 3D, gives a constraint that allows to recover both the motion and the camera's calibration. The constraint is formulated via a set of trifocal tensors, that can be estimated linearly, and from which motion, and then calibration, can be extracted. In [30], this approach is first formulated for the use of 3D calibration objects, and for the general imaging model, i.e. for non-central cameras. We also propose variants of the approach, that may be

important in practice: first, due to the usefulness of planar calibration patterns, we specialized the approach appropriately. Second, we propose a variant that works specifically for central cameras (pinhole, central catadioptric, or any other central camera). More details are given in [29].

5. Pose Estimation

Pose estimation is the problem of computing the relative position and orientation between an object of *known* structure, and a calibrated camera. A literature review on algorithms for pinhole cameras is given in [12]. Here, we briefly show how the minimal case can be solved for general cameras. For pinhole cameras, pose can be estimated, up to a finite number of solutions, from 3 point correspondences (3D-2D) already. The same holds for general cameras. Consider 3 image points and the associated projection rays, computed using the calibration information. We parameterize generic points on the rays as follows: $\mathbf{A}_i + \lambda_i \mathbf{B}_i$.

We know the structure of the observed object, meaning that we know the mutual distances d_{ij} between the 3D points. We can thus write equations on the unknowns λ_i, that parameterize the object's pose:

$$\|\mathbf{A}_i + \lambda_i \mathbf{B}_i - \mathbf{A}_j - \lambda_j \mathbf{B}_j\|^2 = d_{ij}^2 \quad \text{for } (i,j) = (1,2),(1,3),(2,3)$$

This gives a total of 3 equations that are quadratic in 3 unknowns. Many methods exist for solving this problem, e.g. symbolic computation packages such as MAPLE allow to compute a resultant polynomial of degree 8 in a single unknown, that can be numerically solved using any root finding method.

Like for pinhole cameras, there are up to 8 theoretical solutions. For pinhole cameras, at least 4 of them can be eliminated because they would correspond to points lying behind the camera [12]. As for general cameras, determining the maximum number of feasible solutions requires further investigation. In any case, a unique solution can be obtained using one or two additional points [12]. More details on pose estimation for non-central cameras are given in [6, 21].

6. Motion Estimation

We describe how to estimate ego-motion, or, more generally, relative position and orientation of two calibrated general cameras. This is done via a generalization of the classical motion estimation problem for pinhole cameras and its associated centerpiece, the essential matrix [17]. We briefly summarize how the classical problem is usually solved [15]. Let R be the rotation matrix and $\mathbf{t}$ the translation vector describing the motion. The essential matrix is defined as $\mathsf{E} = -[\mathbf{t}]_\times \mathsf{R}$. It can be estimated using point correspondences $(\mathbf{x}_1, \mathbf{x}_2)$ across two views, using the epipolar constraint $\mathbf{x}_2^\mathsf{T} \mathsf{E} \mathbf{x}_1 = 0$. This can be done linearly using 8 correspondences or more. In the minimal case

of 5 correspondences, an efficient non-linear minimal algorithm, which gives exactly the theoretical maximum of 10 feasible solutions, was only recently introduced [20]. Once the essential matrix is estimated, the motion parameters R and $\mathbf{t}$ can be extracted relatively straightforwardly [20].

In the case of our general imaging model, motion estimation is performed similarly, using pixel correspondences $(\mathbf{x}_1, \mathbf{x}_2)$. Using the calibration information, the associated projection rays can be computed. Let them be represented by their Plücker coordinates, i.e. 6-vectors $\mathbf{L}_1$ and $\mathbf{L}_2$. The epipolar constraint extends naturally to rays, and manifests itself by a 6×6 essential matrix, cf. [25] and section 8.3:

$$\mathsf{E} = \begin{pmatrix} -[\mathbf{t}]_\times \mathsf{R} & \mathsf{R} \\ \mathsf{R} & 0 \end{pmatrix}$$

The epipolar constraint then writes: $\mathbf{L}_2^\mathsf{T} \mathsf{E} \mathbf{L}_1 = 0$ [25]. Once E is estimated, motion can again be extracted straightforwardly (e.g., R can simply be read off E). Linear estimation of E requires 17 correspondences.

There is an important difference between motion estimation for central and non-central cameras: with central cameras, the translation component can only be recovered up to scale. Non-central cameras however, allow to determine even the translation's scale. This is because a single calibrated non-central camera already carries scale information (via the distance between mutually skew projection rays). One consequence is that the theoretical minimum number of required correspondences is 6 instead of 5. It might be possible, though very involved, to derive a minimal 6-point method along the lines of [20].

7. 3D Point Triangulation

We now describe an algorithm for 3D reconstruction from two or more calibrated images with known relative position. Let $\mathbf{C} = (X, Y, Z)^\mathsf{T}$ be a 3D point that is to be reconstructed, based on its projections in n images. Using calibration information, we can compute the n associated projection rays. Here, we represent the ith ray using a starting point $\mathbf{A}_i$ and the direction, represented by a unit vector $\mathbf{B}_i$. We apply the mid-point method [14, 25], i.e. determine $\mathbf{C}$ that is closest in average to the n rays. Let us represent generic points on rays using position parameters λ_i. Then, $\mathbf{C}$ is determined by minimizing the following expression over X, Y, Z and the λ_i: $\sum_{i=1}^{n} \|\mathbf{A}_i + \lambda_i \mathbf{B}_i - \mathbf{C}\|^2$.

This is a linear least squares problem, which can be solved e.g. via the Pseudo-Inverse, leading to the following explicit equation (derivations omitted):

$$\begin{pmatrix} \mathbf{C} \\ \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \underbrace{\begin{pmatrix} n\mathtt{I}_3 & -\mathbf{B}_1 & \cdots & -\mathbf{B}_n \\ -\mathbf{B}_1^\mathsf{T} & 1 & & \\ \vdots & & \ddots & \\ -\mathbf{B}_n^\mathsf{T} & & & 1 \end{pmatrix}^{-1}}_{\mathsf{M}} \begin{pmatrix} \mathtt{I}_3 & \cdots & \mathtt{I}_3 \\ -\mathbf{B}_1^\mathsf{T} & & \\ & \ddots & \\ & & -\mathbf{B}_n^\mathsf{T} \end{pmatrix} \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{pmatrix}$$

where $\mathtt{I}_3$ is the identity matrix of size 3×3. Due to its sparse structure, the inversion of the matrix M in this equation, can actually be performed in closed-form. Overall, the triangulation of a 3D point using n rays, can by carried out very efficiently, using only matrix multiplications and the inversion of a symmetric 3×3 matrix (details omitted).

8. Multi-View Geometry

We establish the basics of a multi-view geometry for general (non-central) cameras. Its cornerstones are, as with perspective cameras, matching tensors. We show how to establish them, analogously to the perspective case.

Here, we only talk about the calibrated case; the uncalibrated case is nicely treated for perspective cameras, since calibrated and uncalibrated cameras are linked by projective transformations. For non-central cameras however, there is no such link: in the most general case, every pair (pixel, camera ray) may be completely independent of other pairs.

8.1 Reminder on Multi-View Geometry for Perspective Cameras

We briefly review how to derive multi-view matching relations for perspective cameras [7]. Let P_i be projection matrices and $\mathbf{q}_i$ image points. A set of image points are matching, if there exists a 3D point $\mathbf{Q}$ and scale factors λ_i such that:

$$\lambda_i \mathbf{q}_i = \mathsf{P}_i \mathbf{Q}$$

This may be formulated as the following matrix equation:

$$\underbrace{\begin{pmatrix} \mathsf{P}_1 & \mathbf{q}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathsf{P}_2 & \mathbf{0} & \mathbf{q}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathsf{P}_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{q}_n \end{pmatrix}}_{\mathsf{M}} \begin{pmatrix} \mathbf{Q} \\ -\lambda_1 \\ -\lambda_2 \\ \vdots \\ -\lambda_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

The matrix M, of size $3n \times (4+n)$ has thus a null-vector, meaning that its rank is less than $4+n$. Hence, the determinants of all its submatrices of size

$(4+n)\times(4+n)$ must vanish. These determinants are multi-linear expressions in terms of the coordinates of image points $\mathbf{q}_i$.

They have to be expressed for any possible submatrix. Only submatrices with 2 or more rows per view, give rise to constraints linking all projection matrices. Hence, constraints can be obtained up to n views with $2n \leq 4+n$, meaning that only for up to 4 views, matching constraints linking all views can be obtained.

The constraints for n views take the form:

$$\sum_{i_1=1}^{3}\sum_{i_2=1}^{3}\cdots\sum_{i_n=1}^{3} q_{1,i_1}q_{2,i_2}\cdots q_{n,i_n}T_{i_1,i_2,\cdots,i_n} = 0 \tag{2}$$

where the multi-view matching tensor T of dimension $3\times\cdots\times 3$ depends on and partially encodes the cameras' projection matrices P_i.

Note that as soon as cameras are calibrated, this theory applies to any central camera: for a camera with radial distortion for example, the above formulation holds for distortion-corrected image points.

8.2 Multi-View Geometry for Non-Central Cameras

Here, instead of projection matrices (depending on calibration and pose), we deal with pose matrices:

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

These express the similarity transformations that map a point from some global reference frame, into the camera's local coordinate frames (note that since no optical center and no camera axis exist, no assumptions about the local coordinate frames are made). As for image points, they are now replaced by camera rays. Let the ith ray be represented by two 3D points $\mathbf{A}_i$ and $\mathbf{B}_i$.

Eventually, we will to obtain expressions in terms of the rays' Plücker coordinates, i.e. we will end up with matching tensors T and matching constraints of the form (2), with the difference that tensors will have size $6\times\cdots\times 6$ and act on Plücker line coordinates:

$$\sum_{i_1=1}^{6}\sum_{i_2=1}^{6}\cdots\sum_{i_n=1}^{6} L_{1,i_1}L_{2,i_2}\cdots L_{n,i_n}T_{i_1,i_2,\cdots,i_n} = 0 \tag{3}$$

In the following, we explain how to derive such matching constraints.

Consider a set of n camera rays and let them be defined by two points $\mathbf{A}_i$ and $\mathbf{B}_i$ each; the choice of points to represent a ray is not important, since later we will fall back onto the ray's Plücker coordinates.

Now, a set of n camera rays are matching, if there exist a 3D point $\mathbf{Q}$ and scale factors λ_i and μ_i associated with each ray such that:

$$\lambda_i \mathbf{A}_i + \mu_i \mathbf{B}_i = \mathsf{P}_i \mathbf{Q}$$

i.e. if the point $\mathsf{P}_i\mathbf{Q}$ lies on the line spanned by $\mathbf{A}_i$ and $\mathbf{B}_i$.

Like for perspective cameras, we group these equations in matrix form:

$$\underbrace{\begin{pmatrix} \mathsf{P}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathsf{P}_2 & \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \mathbf{B}_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathsf{P}_n & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_n & \mathbf{B}_n \end{pmatrix}}_{\mathsf{M}} \begin{pmatrix} \mathbf{Q} \\ -\lambda_1 \\ -\mu_1 \\ -\lambda_2 \\ -\mu_2 \\ \vdots \\ -\lambda_n \\ -\mu_n \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}$$

As above, this equation shows that M must be rank-deficient. However, the situation is different here since the P_i are of size 4×4 now, and M of size $4n \times (4 + 2n)$. We thus have to consider submatrices of M of size $(4 + 2n) \times (4 + 2n)$. Furthermore, in the following we show that only submatrices with 3 rows or more per view, give rise to constraints on all pose matrices. Hence, $3n \leq 4 + 2n$, and again, $n \leq 4$, i.e. multi-view constraints are only obtained for up to 4 views.

Let us first see what happens for a submatrix of M where some view contributes only a single row. The two columns corresponding to its base points $\mathbf{A}$ and $\mathbf{B}$, are multiples of one another since they consist of zeroes only, besides a single non-zero coefficient, in the single row associated with the considered view. Hence, the determinant of the considered submatrix of M is always zero, and no constraint is available.

In the following, we exclude this case, i.e. we only consider submatrices of M where each view contributes at least two rows. Let N be such a matrix. Without loss of generality, we start to develop its determinant with the columns containing $\mathbf{A}_1$ and $\mathbf{B}_1$. The determinant is then given as a sum of terms of the following form:

$$(A_{1,j}B_{1,k} - A_{1,k}B_{1,j}) \det \bar{\mathsf{N}}_{jk}$$

where $j, k \in \{1..4\}$, $j \neq k$, and $\bar{\mathsf{N}}_{jk}$ is obtained from N by dropping the columns containing $\mathbf{A}_1$ and $\mathbf{B}_1$ as well as the rows containing $A_{1,j}$ etc.

We observe several things:

- The term $(A_{1,j}B_{1,k} - A_{1,k}B_{1,j})$ is nothing else than one of the Plücker coordinates of the ray of camera 1 (cf. section 2). By continuing with

# cameras	central		non-central	
	M	useful submatrices	M	useful submatrices
2	6×6	3-3	8×8	4-4
3	9×7	3-2-2	12×10	4-3-3
4	12×8	2-2-2-2	16×12	3-3-3-3

Table 2. Cases of multi-view matching constraints for central and non-central cameras. The second columns of "central" and "non-central" contain entries of the form $x - y - z$ etc. This refers to submatrices of M containing x rows from one camera, y from another etc., whose determinant being equal zero, constitutes a matching constraint between all cameras.

the development of the determinant of $\bar{\mathsf{N}}_{jk}$, it becomes clear that the total determinant of N can be written in the form:

$$\sum_{i_1=1}^{6} \sum_{i_2=1}^{6} \cdots \sum_{i_n=1}^{6} L_{1,i_1} L_{2,i_2} \cdots L_{n,i_n} T_{i_1,i_2,\cdots,i_n} = 0$$

i.e. the coefficients of the $\mathbf{A}_i$ and $\mathbf{B}_i$ are "folded together" into the Plücker coordinates of camera rays and T is a matching tensor between the n cameras. Its coefficients depend exactly on the cameras' pose matrices.

- If camera 1 contributes only two rows to N, then the determinant of N becomes of the form:

$$L_{1,x} \left(\sum_{i_2=1}^{6} \cdots \sum_{i_n=1}^{6} L_{2,i_2} \cdots L_{n,i_n} T_{i_2,\cdots,i_n} \right) = 0$$

i.e. it only contains a single coordinate of the ray of camera 1, and the tensor T does not depend at all on the pose of that camera. Hence, to obtain constraints between all cameras, every camera has to contribute at least three rows to the considered submatrix.

We are now ready to establish the different cases that lead to useful multi-view constraints. As mentioned above, for more than 4 cameras, no constraints linking all of them are available: submatrices of size at least $3n \times 3n$ would be needed, but M only has $4 + 2n$ columns. So, only for $n \leq 4$, such submatrices exist.

Table 2 gives all useful cases, both for central and non-central cameras. These lead to two-view, three-view and four-view matching constraints, encoded by essential matrices, trifocal and quadrifocal tensors.

8.3 The Case of Two Views

We have so far explained how to formulate bifocal, trifocal and quadrifocal matching constraints between non-central cameras, expressed via matching

tensors of dimension 6×6 to $6 \times 6 \times 6 \times 6$. To make things more concrete, we explore the two-view case in some more detail in the following. We show how the bifocal matching tensor, or essential matrix, can be expressed in terms of the motion/pose parameters. This is then specialized from non-central to axial cameras.

8.3.1 Non-Central Cameras. For simplicity, we assume here that the global coordinate system coincides with the first camera's local coordinate system, i.e. the first camera's pose matrix is the identity. As for the pose of the second camera, we drop indices, i.e. we express it via a rotation matrix R and a translation vector $\mathbf{t}$. The matrix M is thus given as:

$$\mathsf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & A_{1,1} & B_{1,1} & 0 & 0 \\ 0 & 1 & 0 & 0 & A_{1,2} & B_{1,2} & 0 & 0 \\ 0 & 0 & 1 & 0 & A_{1,3} & B_{1,3} & 0 & 0 \\ 0 & 0 & 0 & 1 & A_{1,4} & B_{1,4} & 0 & 0 \\ R_{11} & R_{12} & R_{13} & t_1 & 0 & 0 & A_{2,1} & B_{2,1} \\ R_{21} & R_{22} & R_{23} & t_2 & 0 & 0 & A_{2,2} & B_{2,2} \\ R_{31} & R_{32} & R_{33} & t_3 & 0 & 0 & A_{2,3} & B_{2,3} \\ 0 & 0 & 0 & 1 & 0 & 0 & A_{2,4} & B_{2,4} \end{pmatrix}$$

For a matching pair of lines, M must be rank-deficient. In this two-view case, this implies that its determinant is equal to zero. As for the determinant, it can be developed to the following expression, where the Plücker coordinates $\mathbf{L}_1$ and $\mathbf{L}_2$ are defined as in equation (1):

$$\mathbf{L}_2^\mathsf{T} \begin{pmatrix} -[\mathbf{t}]_\times \mathsf{R} & \mathsf{R} \\ \mathsf{R} & \mathsf{0} \end{pmatrix} \mathbf{L}_1 = 0 \tag{4}$$

We find the essential matrix E and the epipolar constraint that were already mentioned in section 6.

8.3.2 Axial Cameras. As mentioned in section 3, we adopt local coordinate systems where camera rays have $L_6 = 0$. Hence, the epipolar constraint (4) can be expressed by a reduced essential matrix of size 5×5:

$$\begin{pmatrix} L_{2,1} & \cdots & L_{2,5} \end{pmatrix} \begin{pmatrix} & -[\mathbf{t}]_\times \mathsf{R} & & \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \\ R_{31} & R_{32} \end{pmatrix} \\ \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \end{pmatrix} & & \mathsf{0}_{2\times 2} \end{pmatrix} \begin{pmatrix} L_{1,1} \\ \vdots \\ L_{1,5} \end{pmatrix} = 0$$

Note that this essential matrix is in general of full rank (rank 5), but may be rank-deficient. It can be shown that it is rank-deficient exactly if the two

camera axes cut each other. In that case, the left and right null-vectors of E represent the camera axes of one view in the local coordinate system of the other one (one gets the Plücker vectors when adding a zero between second and third coordinates).

8.3.3 Central Cameras. As mentioned in section 3, we here deal with camera rays of the form $(L_1, L_2, L_3, 0, 0, 0)^\mathsf{T}$. Hence, the epipolar constraint (4) can be expressed by a reduced essential matrix of size 3×3:

$$\begin{pmatrix} L_{2,1} & L_{2,2} & L_{2,3} \end{pmatrix} \begin{pmatrix} -[\mathbf{t}]_\times \mathsf{R} \end{pmatrix} \begin{pmatrix} L_{1,1} \\ L_{1,2} \\ L_{1,3} \end{pmatrix} = 0$$

We actually find here the "classical" 3×3 essential matrix $-[\mathbf{t}]_\times \mathsf{R}$ [15, 17].

9. Experimental Results

We describe a few experiments on calibration, motion estimation and 3D reconstruction, on the following three indoor scenarios:

- A house scene, captured by an omnidirectional camera and a stereo system.
- A house scene, captured by an omnidirectional and a pinhole camera.
- A scene consisting of a set of objects placed in random positions as shown in Figure 3(b), captured by an omnidirectional and a pinhole camera.

9.1 Calibration

We calibrate three types of cameras here: pinhole, stereo, and omni-directional systems.

Pinhole Camera: Figure 2(a) shows the calibration of a pinhole camera using the single center assumption [30].

Stereo camera: Here we calibrate the left and right cameras separately as two individual pinhole cameras. In the second step we capture an image of a same scene from left and right cameras and compute the motion between them using the technique described in section 6. Finally using the computed motion we obtain both the rays of left camera and the right camera in the same coordinate system, which essentially provides the required calibration information.

Omni-directional camera: Our omni-directional camera is a Nikon Coolpix-5400 camera with an E-8 Fish-Eye lens. Its field of view is 360×183. In theory, this is just another pinhole camera with large distortions. The calibration results are shown in Figure 2. Note that we have calibrated only a part of

the image because three images are insufficient to capture the whole image in an omnidirectional camera. By using more than three boards it is possible to cover the whole image.

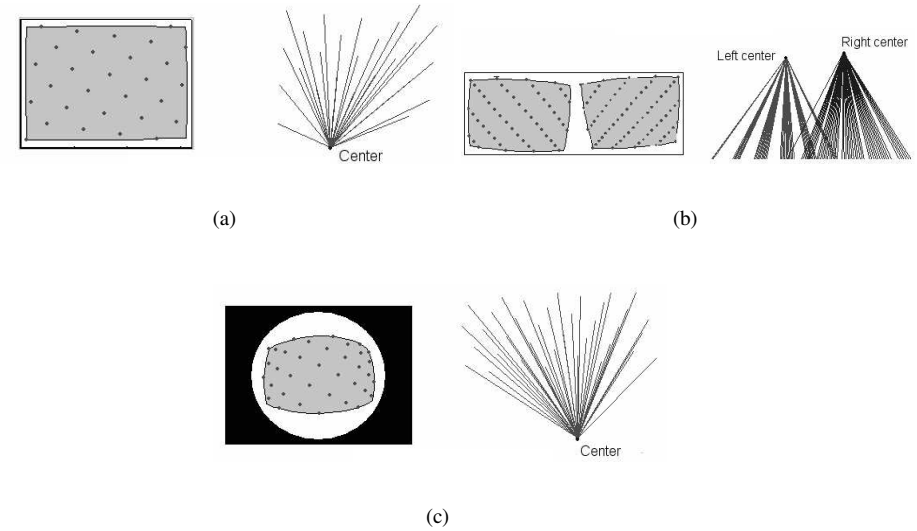
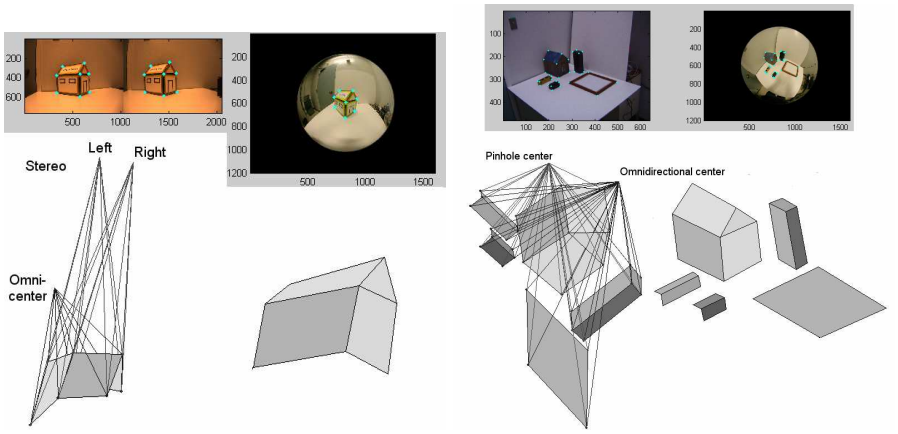


Figure 2. (a) Pinhole. (b) Stereo. (c) Omni-directional (fish-eye). The shading shows the calibrated region and the 3D rays on the right correspond to marked image pixels.

9.2 Motion and Structure Recovery

Pinhole and Omni-directional: Pinhole and omni-directional cameras are both central. Since the omni-directional camera has a very large field of view and consequently lower resolution compared to pinhole camera, the images taken from close viewpoints from these two cameras have different resolutions as shown in Figure 3. This poses a problem in finding correspondences between keypoints. Operators like SIFT [18], which are scale invariant, are not camera invariant. Direct application of SIFT failed to provide good results in our scenario. Thus we had to manually give the correspondences. One interesting research direction would be to work on the automatic matching of feature points in these images.

Stereo system and Omni-directional: A stereo system can be considered as a non-central camera with two centers. The image of a stereo system is a concatenated version of left and right camera images. Therefore the same scene point appears more than once in the image. While finding image correspondences one keypoint in the omni-directional image may correspond to



(a) (b)

Figure 3. (a) Stereo and omni-directional. (b) Pinhole and omni-directional. We intersect the rays corresponding to the matching pixels in the images to compute the 3D points.

2 keypoints in the stereo system as shown in Figure 3(a). Therefore in the ray-intersection we intersect three rays to find one 3D point.

10. Conclusion

We have reviewed calibration and structure from motion tasks for the general non-central camera model. We also proposed a multi-view geometry for non-central cameras. A natural hierarchy of camera models has been introduced, grouping cameras into classes depending on, loosely speaking, the spatial distribution of their projection rays.

Among ongoing and future works, there is the adaptation of our calibration approach to axial and other camera models. We also continue our work on bundle adjustment for the general imaging model, cf. [26], and the exploration of hybrid systems, combining cameras of different types [28, 26].

Acknowledgements. This work was partially supported by the NSF grant ACI-0222900 and by the Multidisciplinary Research Initiative (MURI) grant by Army Research Office under contract DAA19-00-1-0352.

References

[1] S. Baker and S.K. Nayar. A Theory of Single-Viewpoint Catadioptric Image Formation. *IJCV*, 35(2), pp. 1-22, 1999.

[2] H. Bakstein. Non-central cameras for 3D reconstruction. Technical Report CTU-CMP-2001-21, Center for Machine Perception, Czech Technical University, Prague, 2001.

[3] H. Bakstein and T. Pajdla. An overview of non-central cameras. *Computer Vision Winter Workshop*, Ljubljana, Slovenia, pp. 223-233, 2001.

[4] J. Barreto and H. Araujo. Paracatadioptric Camera Calibration Using Lines. *International Conference on Computer Vision*, Nice France, pp. 1359-1365, 2003.

[5] G. Champleboux, S. Lavallée, P. Sautot and P. Cinquin. Accurate Calibration of Cameras and Range Imaging Sensors: the NPBS Method. *International Conference on Robotics and Automation*, Nice, France, pp. 1552-1558, 1992.

[6] C.-S. Chen and W.-Y. Chang. On Pose Recovery for Generalized Visual Sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7), pp. 848-861, 2004.

[7] O. Faugeras and B. Mourrain. On the Geometry and Algebra of the Point and Line Correspondences Between N Images. *International Conference on Computer Vision*, Cambridge, MA, USA, pp. 951-956, 1995.

[8] C. Geyer and K. Daniilidis. A unifying theory of central panoramic systems and practical applications. *European Conference on Computer Vision*, Dublin, Ireland, Vol. II, pp. 445-461, 2000.

[9] C. Geyer and K. Daniilidis. Paracatadioptric camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), pp. 687-695, 2002.

[10] K.D. Gremban, C.E. Thorpe and T. Kanade. Geometric Camera Calibration using Systems of Linear Equations. *International Conference on Robotics and Automation*, Philadelphia, USA, pp. 562-567, 1988.

[11] M.D. Grossberg and S.K. Nayar. A general imaging model and a method for finding its parameters. *International Conference on Computer Vision*, Vancouver, Canada, Vol. 2, pp. 108-115, 2001.

[12] R.M. Haralick, C.N. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3), pp. 331-356, 1994.

[13] R.I. Hartley and R. Gupta. Linear Pushbroom Cameras. *European Conference on Computer Vision*, Stockholm, Sweden, pp. 555-566, 1994.

[14] R.I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2), pp. 146-157, 1997.

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

[16] R.A. Hicks and R. Bajcsy. Catadioptric Sensors that Approximate Wide-angle Perspective Projections. *Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, USA, pp. 545-551, 2000.

[17] H.C. Longuet-Higgins. A Computer Program for Reconstructing a Scene from Two Projections. *Nature*, 293, pp. 133-135, 1981.

[18] D.G. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, Kerkyra, Greece, pp. 1150-1157, 1999.

[19] J. Neumann, C. Fermüller, and Y. Aloimonos. Polydioptric Camera Design and 3D Motion Estimation. *Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, Vol. II, pp. 294-301, 2003.

[20] D. Nistér. An Efficient Solution to the Five-Point Relative Pose Problem. *Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, Vol. II, pp. 195-202, 2003.

[21] D. Nistér. A Minimal Solution to the Generalized 3-Point Pose Problem. *Conference on Computer Vision and Pattern Recognition*, Washington DC, USA, Vol. 1, pp. 560-567, 2004.

[22] T. Pajdla. Geometry of Two-Slit Camera. Technical Report CTU-CMP-2002-02, Center for Machine Perception, Czech Technical University, Prague, 2002.

[23] T. Pajdla. Stereo with oblique cameras. *International Journal of Computer Vision*, 47(1), pp. 161-170, 2002.

[24] S. Peleg, M. Ben-Ezra, Y. Pritch. OmniStereo: Panoramic Stereo Imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3), pp. 279-290, 2001.

[25] R. Pless. Using Many Cameras as One. *Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, Vol. II, pp. 587-593, 2003.

[26] S. Ramalingam, S. Lodha and P. Sturm. A Generic Structure-from-Motion Algorithm for Cross-Camera Scenarios. *5th Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, Prague, Czech Republic, pp. 175-186, 2004.

[27] H.-Y. Shum, A. Kalai, S.M. Seitz. Omnivergent Stereo. *International Conference on Computer Vision*, Kerkyra, Greece, pp. 22-29, 1999.

[28] P. Sturm. Mixing catadioptric and perspective cameras. *Workshop on Omnidirectional Vision*, Copenhagen, Denmark, pp. 60-67, 2002.

[29] P. Sturm and S. Ramalingam. A generic calibration concept-theory and algorithms. Research Report 5058, INRIA, 2003.

[30] P. Sturm and S. Ramalingam. A generic concept for camera calibration. *European Conference on Computer Vision*, Prague, Czech Republic, pp. 1-13, 2004.

[31] R. Swaminathan, M.D. Grossberg, and S.K. Nayar. A perspective on distortions. *Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, Vol. II, pp. 594-601, 2003.

[32] J. Yu and L. McMillan. General Linear Cameras. *European Conference on Computer Vision*, Prague, Czech Republic, pp. 14-27, 2004.

[33] A. Zomet, D. Feldman, S. Peleg and D. Weinshall. Mosaicing New Views: The Crossed-Slit Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6), pp. 741-754, 2003.