

# Affine Structure and Motion from Points, Lines and Conics

FREDRIK KAHL\*, ANDERS HEYDEN\*\*

{fredrik,heyden}@maths.lth.se

*Centre for Mathematical Sciences, Lund University, Box 118, S-221 00 Lund, Sweden*

;

Editors: ??

**Abstract.** In this paper several new methods for estimating scene structure and camera motion from an image sequence taken by affine cameras are presented. All methods can incorporate both point, line and conic features in a unified manner. The correspondence between features in different images is assumed to be known.

Three new tensor representations are introduced describing the viewing geometry for two and three cameras. The centred affine epipoles can be used to constrain the location of corresponding points and conics in two images. The third order, or alternatively, the reduced third order centred affine tensors can be used to constrain the locations of corresponding points, lines and conics in three images. The reduced third order tensors contain only 12 components compared to the 16 components obtained when reducing the trifocal tensor to affine cameras.

A new factorization method is presented. The novelty lies in the ability to handle not only point features, but also line and conic features concurrently. Another complementary method based on the so-called closure constraints is also presented. The advantage of this method is the ability to handle missing data in a simple and uniform manner. Finally, experiments performed on both simulated and real data are given, including a comparison with other methods.

Keywords: Reconstruction, Affine cameras, Matching constraints, Closure constraints, Factorization methods, Multiple view tensors

## 1. Introduction

Reconstruction of a three-dimensional object from a number of its two-dimensional images is one of the core problems in computer vision. Both the structure of the object and the motion of the camera are assumed to be unknown. Many approaches have been proposed to this problem and apart

from the reconstructed object also the camera motion is obtained, cf. (Tomasi and Kanade 1992, Koenderink and van Doorn 1991, McLauchlan and Murray 1995, Sturm and Triggs 1996, Sparr 1996, Shashua and Navab 1996, Weng, Huang and Ahuja 1992, Ma 1993).

There are two major difficulties that have to be dealt with. The first one is to obtain corresponding points (or lines, conics, etc.) throughout the sequence. The second one is to choose an appropriate camera model, e.g., perspective (calibrated or uncalibrated), weak perspective, affine, etc. Moreover, these two problems are not com-

\*Supported by the ESPRIT Reactive LTR project 21914, CUMULI

\*\*Supported by the Swedish Research Council for Engineering Sciences (TFR), project 95-64-222

pletely separated, but in some sense coupled to each other, which will be explained in more detail later.

The first problem of obtaining feature correspondences between different images is simplified if the viewing positions are close together. However, most reconstruction algorithms break down when the viewpoints are close together, especially in the perspective case. The correspondence problem is not addressed here. Instead we assume that the correspondences are known.

The problem of choosing an appropriate camera model is somewhat complex. If the intrinsic parameters of the camera are known, it seems reasonable to choose the calibrated perspective (pinhole) camera, see (Maybank 1993). If the intrinsic parameters are unknown, many researchers have proposed the uncalibrated perspective (projective) camera, see (Faugeras 1992). This is the most appealing choice from a theoretical point of view, but in practice it has a lot of drawbacks. Firstly, only the projective structure of the scene is recovered, which is often not sufficient. Secondly, the images have to be captured from widespread locations, with large perspective effects, which is rarely the case if the imaging situation cannot be completely controlled. If this condition is not fulfilled, the reconstruction algorithm may give a very inaccurate result and might even break down completely. Thirdly, the projective group is in some sense too large for practical applications. Theoretically, the projective group is the correct choice, but only a small part of the group is actually relevant for most practical situations, leading to too many degrees of freedom in the model.

Another proposed camera model is the affine one, see (Mundy and Zisserman 1992), which is an approximation of the perspective camera model. This is the model that will be used in this paper. The advantages of using the affine camera model, compared to the perspective one, are many-fold. Firstly, the affine structure of the scene is obtained instead of the projective in the uncalibrated perspective case. Secondly, the images may be captured from nearby locations without the algorithms breaking down. Again, this facilitates the correspondence problem. Thirdly, the geometry and algebra are more simple, leading to more efficient and robust reconstruction algorithms. Also, there is a lack of satisfactory algorithms for non-

point features in the perspective case, especially for conics and curves.

This paper presents an integrated approach to the structure and motion problem for affine cameras. We extend current approaches to affine structure and motion in several directions, cf. (Tomasi and Kanade 1992, Shapiro, Zisserman and Brady 1995, Quan and Kanade 1997, Koenderink and van Doorn 1991). One popular reconstruction method for affine cameras is the Tomasi-Kanade factorization method for point correspondences, see (Tomasi and Kanade 1992). We will generalize the factorization idea to be able to incorporate also corresponding lines and conics. In (Quan and Kanade 1997) a line-based factorization method is presented and in (Triggs 1996) a factorization algorithm for both points and lines in the projective case is given.

Another approach to reconstruction from images is to use the so-called matching constraints. These constraints are polynomial expressions in the image coordinates and they constrain the locations of corresponding features in two, three or four images, see (Triggs 1997, Heyden 1995) for a thorough treatment in the projective case. The drawback of using matching constraints is that only two, three or four images can be used at the same time. The advantage is that missing data, e.g. a point that is not visible in all images, can be handled automatically. In this paper the corresponding matching constraints for the affine camera in two and three images are derived. Specializing the projective matching constraints directly, like in (Torr 1995), will lead to a large overparameterization. We will not follow this path, instead the properties of the affine camera will be taken into account and a more effective parameterization is obtained. It is also shown how to concatenate these constraints in a unified manner to be able to cope with sequences of images. This will be done using the so-called closure constraints, constraining the coefficients of the matching constraints and the camera matrices. Similar constraints have been developed in the projective case, see (Triggs 1997). Some attempts to deal with the missing data problem have been made in (Tomasi and Kanade 1992, Jacobs 1997). We describe these methods and the relationship to our approach based on closure constraints, and we also

provide an experimental comparison with Jacobs' method.

Preliminary results of this work, primarily based on the matching constraints for image triplets and the factorization method can be found in (Kahl and Heyden 1998). Recently, the matching constraints for two and three affine views have also been derived in a similar manner, but independently, in two other papers. In (Bretzner and Lindeberg 1998), the projective trifocal tensor is first specialized to the affine case, like in (Torr 1995), resulting in 16 non-zero coefficients in the trifocal tensor. Then, they introduce the centred affine trifocal tensor by using relative coordinates, reducing the number of coefficients to 12. From these representations, they calculate the three orthographic camera matrices corresponding to these views in a rather complicated way. A factorization method for points and lines for longer sequences is also developed. In (Quan and Ohta 1998) the two-view and three-view constraints are derived in a nice and compact way for centred affine cameras. By examining the relationships between the two- and three-view constraints, they are able to reduce the number of coefficients to only 10 for the three-view case. These 10 coefficients for three affine cameras are then directly related to the parameters of three orthographic cameras. Our presentation of the matching constraints is similar to the one in (Quan and Ohta 1998), but we prefer to use a tensorial notation. While we pursue the path of coping with longer image sequences, their work is more focused on obtaining a Euclidean reconstruction limited to three calibrated cameras.

The paper is organized as follows. In Section 2, we give a brief review of the affine camera, describing how points, lines and conics project onto the image plane. In Section 3, the matching constraints for two and three views are described. For arbitrary many views, two alternative approaches are presented. The first one, in Section 4, is based on factorization and the second one, in Section 5, is based on closure constraints that can handle missing data. In Section 5, we also describe two related methods to the missing data problem. A number of experiments, performed on both simulated and on real data, is presented in Section 6. Finally, in Section 7, some conclusions are given.

## 2. The affine camera model

In this section we give a brief review of the affine camera model and describe how different points, lines and quadrics are projected onto the image plane. For a more thorough treatment, see (Shapiro 1995) for points and (Quan and Kanade 1997) for lines.

The projective/perspective camera is modeled by

$$\lambda \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = P \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}, \quad \lambda \neq 0, \quad (1)$$

where  $P$  denotes the standard  $3 \times 4$  camera matrix and  $\lambda$  a scale factor. Here  $\mathbf{X}$  is a 3-vector and  $\mathbf{x}$  is a 2-vector, denoting point coordinates in the 3D scene and in the image respectively.

The affine camera model, first introduced by Mundy and Zisserman in (Mundy and Zisserman 1992), has the same form as (1), but the camera matrix is restricted to

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & p_{34} \end{bmatrix} \quad (2)$$

and the homogeneous scale factor  $\lambda$  is the same for all points. It is an approximation of the projective camera and it generalizes the orthographic, the weak perspective and the para-perspective camera models. These models provide a good approximation of the projective camera when the distances between different points of the object is small compared to the viewing distance. The affine camera has eight degrees of freedom, since (2) is only defined up to a scale factor, and it can be seen as a projective camera with its optical centre on the plane at infinity.

Rewriting the camera equation (1) with the affine restriction (2), the equation can be written

$$\mathbf{x} = A\mathbf{X} + b, \quad (3)$$

where

$$A = \frac{1}{p_{34}} \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{bmatrix} \quad \text{and} \quad b = \frac{1}{p_{34}} \begin{bmatrix} p_{14} \\ p_{24} \end{bmatrix}.$$

A simplification can be obtained by using relative coordinates with respect to some reference point,  $\mathbf{X}_0$ , in the object and the corresponding point  $\mathbf{x}_0 = A\mathbf{X}_0 + b$  in the image. Introducing the relative coordinates  $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0$  and

$\Delta\mathbf{X} = \mathbf{X} - \mathbf{X}_0$ , (3) simplifies to

$$\Delta\mathbf{x} = A\Delta\mathbf{X} . \quad (4)$$

In the following, the reference point will be chosen as the centroid of the point configuration, since the centroid of the three-dimensional point configuration projects onto the centroid of the two-dimensional point configuration. Notice that the visible point configuration may differ from view to view and thus the centroid changes from view to view. This must be considered and we will comment upon it later.

A line in the scene through a point  $\mathbf{X}$  with direction  $\mathbf{D}$  can be written  $\mathbf{L} = \mathbf{X} + \mu\mathbf{D}$ ,  $\mu \in \mathbb{R}$ . With the affine camera, this line is projected to the image line,  $\mathbf{l}$ , through the point  $\mathbf{x} = A\mathbf{X} + b$  according to

$$\begin{aligned} \mathbf{l} &= A\mathbf{L} + b = A(\mathbf{X} + \mu\mathbf{D}) + b = \\ &= A\mathbf{X} + \mu A\mathbf{D} + b = \mathbf{x} + \mu A\mathbf{D} . \end{aligned} \quad (5)$$

Thus, it follows that the direction,  $\mathbf{d}$ , of the image line is obtained as

$$\lambda\mathbf{d} = A\mathbf{D}, \quad \lambda \in \mathbb{R} . \quad (6)$$

This observation was first made in (Quan and Kanade 1997). Notice that the only difference between the projection of points in (4) and the projection of directions of lines in (6) is the scale factor  $\lambda$  present in (6), but not in (4). Thus, with known scale factor  $\lambda$ , a direction can be treated as an ordinary point. This fact will be used later on in the factorization algorithm.

For conics, the situation is a little more complicated than for points and lines. A general conic curve in the plane can be represented by its dual form, the conic envelope,

$$\mathbf{u}^T l \mathbf{u} = 0 , \quad (7)$$

where  $l$  denotes a  $3 \times 3$  symmetric matrix and  $\mathbf{u} = [u \ v \ 1]^T$  denotes extended dual coordinates in the image plane. In the same way, a general quadric surface in the scene can be represented by its dual form, the quadric envelope,

$$\mathbf{U}^T L \mathbf{U} = 0 , \quad (8)$$

where  $L$  denotes a  $4 \times 4$  symmetric matrix and  $\mathbf{U} = [U \ V \ W \ 1]^T$  denotes extended dual coordinates in the 3D space. A conic or a quadric, (7)

or (8), is said to be proper if its matrix is non-singular, otherwise it is said to be degenerate. For most practical situations, it is sufficient to know that a quadric envelope degenerates into a disc quadric, i.e., a conic lying in a plane in space. For more details, see (Semple and Kneebone 1952). The image, under a perspective projection, of a quadric,  $L$ , is a conic,  $l$ . This relation is expressed by

$$\lambda l = P L P^T , \quad (9)$$

where  $P$  is the camera matrix and  $\lambda$  a scale factor. Introducing

$$l = \begin{bmatrix} l_1 & l_2 & l_4 \\ l_2 & l_3 & l_5 \\ l_4 & l_5 & l_6 \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} L_1 & L_2 & L_4 & L_7 \\ L_2 & L_3 & L_5 & L_8 \\ L_4 & L_5 & L_6 & L_9 \\ L_7 & L_8 & L_9 & L_{10} \end{bmatrix}$$

and specializing (9) to the affine camera (3) gives two set of equations. The first set is

$$\begin{aligned} \lambda \begin{bmatrix} l_1 & l_2 \\ l_2 & l_3 \end{bmatrix} &= A \begin{bmatrix} L_1 & L_2 & L_4 \\ L_2 & L_3 & L_5 \\ L_4 & L_5 & L_6 \end{bmatrix} A^T + \\ &+ A [L_7 \ L_8 \ L_9]^T b^T + \\ &+ b [L_7 \ L_8 \ L_9] A^T + b L_{10} b^T , \end{aligned} \quad (10)$$

containing three non-linear equations in  $A$  and  $b$ . Normalizing  $l$  such that  $l_6 = 1$  and  $L$  such that  $L_{10} = 1$ , the second set becomes

$$\begin{bmatrix} l_4 \\ l_5 \end{bmatrix} = A \begin{bmatrix} L_7 \\ L_8 \\ L_9 \end{bmatrix} + b , \quad (11)$$

containing three linear equations in  $A$  and  $b$ . Observe that this equation is of the same form as (3), which implies that conics can be treated in the same way as points, when the non-linear equations in (10) are omitted.

The geometrical interpretation of (11) is that the centre of the quadric projects onto the centre of the conic in the image, since indeed  $[l_4/l_6 \ l_5/l_6]^T$  corresponds to the centre of the conic. This can be seen by parameterizing the conic by its centre point then expressing it in the form of (7).

### 3. Affine matching constraints

The matching constraints in the projective case are well-known and they can directly be specialized to the affine case, cf. (Torr 1995). However, we will not follow this path. Instead, we start from the affine camera equation in (4) leading to fewer parameters and thereby a more effective way of parameterizing the matching constraints.

We will from now on assume that relative coordinates have been chosen and use the notation

$$\mathbf{x}_I = [\mathbf{x}_I^1 \ \mathbf{x}_I^2]^T ,$$

for relative coordinates. The subindex indicates that the image point belongs to image  $I$ .

#### 3.1. Two-view constraints

Denote the two camera matrices corresponding to views number  $I$  and  $J$  by  $A_I$  and  $A_J$  and an arbitrary 3D-point by  $\mathbf{X}$  (in relative coordinates). Then (4) gives for these two images  $\mathbf{x}_I = A_I \mathbf{X}$  and  $\mathbf{x}_J = A_J \mathbf{X}$ , or equivalently,

$$M \begin{bmatrix} \mathbf{X} \\ -1 \end{bmatrix} = \begin{bmatrix} A_I & \mathbf{x}_I \\ A_J & \mathbf{x}_J \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ -1 \end{bmatrix} = 0 .$$

Thus, it follows that  $\det M = 0$  since  $M$  has a non-trivial nullspace. Expanding the determinant by the last column gives one linear equation in the image coordinates  $\mathbf{x}_I = [\mathbf{x}_I^1 \ \mathbf{x}_I^2]^T$  and  $\mathbf{x}_J = [\mathbf{x}_J^1 \ \mathbf{x}_J^2]^T$ . The coefficients of this linear equation depend only on the camera matrices  $A_I$  and  $A_J$ . Therefore, let

$$E_{IJ} = \begin{bmatrix} A_I \\ A_J \end{bmatrix} . \quad (12)$$

**Definition 1.** The minors built up by three different rows from  $E_{IJ}$  in (12) will be called the centred affine epipoles and its 4 components will be denoted by  $\mathbf{E}_{IJ} = (IJ\mathbf{e}^i, JI\mathbf{e}^j)$ , where  $i, j = 1, 2$  and

$$IJ\mathbf{e}^i = \det \begin{bmatrix} A_I^i \\ A_J \end{bmatrix} \quad \text{and} \quad JI\mathbf{e}^j = \det \begin{bmatrix} A_J^j \\ A_I \end{bmatrix} ,$$

where  $A_I^i$  denotes the  $i$ th row of  $A_I$  and similarly for  $A_J$ .

**Remark.** The vector  $_{IJ}\mathbf{e} = (_{IJ}\mathbf{e}^1, _{IJ}\mathbf{e}^2)$  is the well-known epipole or epipolar direction, i.e., the projection in camera  $I$  of the focal point corresponding to camera  $J$ . Here the focal point is a point on the plane at infinity, corresponding to the direction of projection.

Observe that  $\mathbf{E}_{IJ}$  is built up by two different tensors,  $_{IJ}\mathbf{e}^i$  and  $_{JI}\mathbf{e}^j$ , which are contravariant tensors. This terminology alludes to the transformation properties of the tensor components. In fact, consider a change of image coordinates from  $\mathbf{x}$  to  $\hat{\mathbf{x}}$  according to

$$\mathbf{x} = S\hat{\mathbf{x}} \quad \text{or equivalently} \quad \mathbf{x}^i = s_{i'}^i \hat{\mathbf{x}}^{i'} , \quad (13)$$

where  $S$  denotes a non-singular  $2 \times 2$  matrix and  $s_{i'}^i$  denotes  $(S)_{i',i}$ , i.e., the element with row-index  $i$  and column-index  $i'$  of  $S$ . Then the tensor components change according to

$$\mathbf{e}^i = s_{i'}^i \hat{\mathbf{e}}^{i'} .$$

Observe that Einstein's summation convention has been used, i.e., when an index appears twice in a formula it is assumed that a summation is made over that index.

Using this notation the two-view constraint can be written in tensor form as

$$\epsilon_{jj'} JI\mathbf{e}^j \mathbf{x}_J^{j'} + \epsilon_{ii'} IJ\mathbf{e}^i \mathbf{x}_I^{i'} = 0 , \quad (14)$$

where  $\epsilon_{jj'}$  denotes the permutation symbol, i.e.,  $\epsilon_{11} = \epsilon_{22} = 0$ ,  $\epsilon_{12} = 1$  and  $\epsilon_{21} = -1$ . Using instead vector notations the constraint can be written as

$$_{IJ}\mathbf{e} \wedge \mathbf{x}_I + _{JI}\mathbf{e} \wedge \mathbf{x}_J = 0 ,$$

where  $\wedge$  denote the 2-component cross product, i.e.,  $(x_1, x_2) \wedge (y_1, y_2) = x_1 y_2 - x_2 y_1$ . Writing out (14) explicitly gives

$$_{JI}\mathbf{e}^1 \mathbf{x}_J^2 - _{JI}\mathbf{e}^2 \mathbf{x}_J^1 + _{IJ}\mathbf{e}^1 \mathbf{x}_I^2 - _{IJ}\mathbf{e}^2 \mathbf{x}_I^1 = 0 .$$

**Remark.** The tensors could equivalently have been defined as

$$_{IJ}\mathbf{e}_i = \epsilon_{ii'} \det \begin{bmatrix} A_I^{i'} \\ A_J \end{bmatrix}$$

giving a covariant tensor instead. The relations between these tensors are  $_{IJ}\mathbf{e}_i = \epsilon_{ii'} _{IJ}\mathbf{e}^{i'}$ ,  $_{IJ}\mathbf{e}^i =$

$-\epsilon^{ii'} {}_{IJ}\mathbf{e}_{i'}$  and  ${}_{IJ}\mathbf{e}_i {}_{IJ}\mathbf{e}^i = 0$ . The two-view constraints can now simply be written, using the covariant epipolar tensors, as

$${}_{IJ}\mathbf{e}_i \mathbf{x}_I^i + {}_{JI}\mathbf{e}_j \mathbf{x}_J^j = 0 .$$

The choice of covariant or contravariant indices for these 2D tensors is merely a matter of taste. The choice made here to use the contravariant tensors is done because they have physical interpretations as epipoles.

The four components of the centred affine epipoles can be estimated linearly from at least four point or conic correspondences in the two images. In fact, each corresponding feature gives one linear constraint on the components and the use of relative coordinates makes one constraint linearly dependent on the other ones. Corresponding lines in only two views do not constrain the camera motion. From (14) follows that the components can only be determined up to scale. This means that if  $\mathbf{E}_{IJ} = ({}_{IJ}\mathbf{e}^i, {}_{JI}\mathbf{e}^j)$  are centred affine epipoles, then  $\lambda\mathbf{E}_{IJ} = (\lambda{}_{IJ}\mathbf{e}^i, \lambda{}_{JI}\mathbf{e}^j)$ , where  $0 \neq \lambda \in \mathbb{R}$ , are also centred affine epipoles corresponding to the same viewing geometry. This undetermined scale factor corresponds to the possibility to rescale both the reconstruction and the camera matrices, keeping (4) valid.

The tensor components parameterize the epipolar geometry in two views. However, the camera matrices are only determined up to an unknown affine transformation. One possible choice of camera matrices is given by the following proposition.

**Proposition 1.** Given centred affine epipoles,  $\mathbf{E}_{IJ} = ({}_{IJ}\mathbf{e}^i, {}_{JI}\mathbf{e}^j)$  normalized such that  ${}_{JI}\mathbf{e}^1 = 1$ , a set of corresponding camera matrices is given by

$$A_I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

$$A_J = \begin{bmatrix} 0 & 0 & 1 \\ {}_{IJ}\mathbf{e}^2 & -{}_{IJ}\mathbf{e}^1 & {}_{JI}\mathbf{e}^2 \end{bmatrix} .$$

**Proof:** The result follows from straightforward calculations of the minors in (12).  $\square$

### 3.2. Three-view constraints

Denote the three camera matrices corresponding to views number  $I$ ,  $J$  and  $K$  by  $A_I$ ,  $A_J$  and  $A_K$  and an arbitrary 3D-point by  $\mathbf{X}$ . Then, the projection of  $\mathbf{X}$  (in relative coordinates) in these images are given by  $\mathbf{x}_I = A_I \mathbf{X}$ ,  $\mathbf{x}_J = A_J \mathbf{X}$  and  $\mathbf{x}_K = A_K \mathbf{X}$  according to (4), or equivalently

$$M \begin{bmatrix} \mathbf{X} \\ -1 \end{bmatrix} = \begin{bmatrix} A_I & \mathbf{x}_I \\ A_J & \mathbf{x}_J \\ A_K & \mathbf{x}_K \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ -1 \end{bmatrix} = 0 . \quad (15)$$

Thus, it follows that  $\text{rank } M < 4$  since  $M$  has a non-trivial nullspace. This means that all  $4 \times 4$  minors of  $M$  vanish. There are in total  $\binom{6}{4} = 15$  such minors and expanding these minors by the last column gives linear equations in the image coordinates  $\mathbf{x}_I$ ,  $\mathbf{x}_J$  and  $\mathbf{x}_K$ . The coefficients of these linear equations are minors formed by three rows from the camera matrices  $A_I$ ,  $A_J$  and  $A_K$ . Let

$$T_{IJK} = \begin{bmatrix} A_I \\ A_J \\ A_K \end{bmatrix} . \quad (16)$$

The minors from (16) are the Grassman coordinates of the linear subspace of  $\mathbb{R}^6$  spanned by the columns of  $T_{IJK}$ . We will use a slightly different terminology and notation, according to the following definition.

**Definition 2.** The  $\binom{6}{3} = 20$  determinants of the matrices built up by three rows from  $T_{IJK}$  in (16) will be denoted by  $\mathbf{T}_{IJK} = (\mathbf{t}^{ijk}, {}_{IJ}\mathbf{e}^i, {}_{IK}\mathbf{e}^i, {}_{JI}\mathbf{e}^j, {}_{JK}\mathbf{e}^j, {}_{KI}\mathbf{e}^k, {}_{KJ}\mathbf{e}^k)$ , where  $\mathbf{e}$  denotes the previously defined centred affine epipoles and  $\mathbf{t}^{ijk}$  will be called the centred affine tensor defined by

$$\mathbf{t}^{ijk} = \det \begin{bmatrix} A_I^i \\ A_J^j \\ A_K^k \end{bmatrix} , \quad (17)$$

where  $A_I^i$  again denotes the  $i$ th row of  $A_I$  and all indices  $i$ ,  $j$  and  $k$  range from 1 to 2.

Observe that  $\mathbf{T}_{IJK}$  is built up by 7 different tensors, the 6 centred affine epipoles,  ${}_{IJ}\mathbf{e}^i$ , etc., and a third order tensor  $\mathbf{t}^{ijk}$ , which is contravariant in all indices<sup>1</sup>. This third order tensor transforms

according to

$$\mathbf{t}^{ijk} = s_{i'}^i u_{j'}^j v_{k'}^k \mathbf{t}^{i'j'k'} ,$$

when coordinates in the images are changed according to (13) in image  $I$  and similarly for image  $J$  and  $K$  using matrices  $U$  and  $V$  instead of  $S$ .

Given point coordinates in all three images, the minors obtained from  $M$  in (15) yield linear constraints on the 20 numbers in the centred affine tensors. One example of such a linear equation, obtained by picking the first, second, third and fifth row of  $M$  is

$$JI\mathbf{e}^1 x_K^1 - KI\mathbf{e}^1 x_J^1 + \mathbf{t}^{111} x_I^2 - \mathbf{t}^{211} x_I^1 = 0 .$$

The general form of such a constraint is

$$\epsilon_{ii'} \mathbf{t}^{ijk} x_I^{i'} - KI\mathbf{e}^k x_J^j + JI\mathbf{e}^j x_K^k = 0 \quad (18)$$

or

$$\epsilon_{jj'} IJ\mathbf{e}^j \mathbf{x}_J^{j'} + \epsilon_{ii'} JI\mathbf{e}^i \mathbf{x}_I^{i'} = 0 , \quad (19)$$

where the last equation is the previously defined two-view constraint. In (18),  $j$  and  $k$  can be chosen in 4 different ways and the different images can be permuted in 3 ways, so there are 12 linear constraints from this equation. Adding the 3 additional two-view constraints from (19) gives in total 15 linear constraints on the 20 tensor components. All constraints can be written

$$Rt = 0 , \quad (20)$$

where  $R$  is a  $15 \times 20$  matrix containing relative image coordinates of the image point and  $t$  is a vector containing the 20 components of the centred affine tensor. From (20), it follows that the overall scale of the tensor components cannot be determined. Observe that since relative coordinates are used, one point alone gives no constraints on the tensor components, since its relative coordinates are all zero. The number of linearly independent constraints for different number of point correspondences is given by the following proposition.

**Proposition 2.** Two corresponding points in 3 images give in general 10 linearly independent constraints on the components of  $\mathbf{T}_{IJK}$ . Three points give in general 16 constraints and four or more points give in general 19 constraints. Thus the centred affine tensor and the centred affine epipoles can in general be linearly recovered from at least four point correspondences in 3 images.

**Proof:** See Appendix A.  $\square$

The next question is how to calculate the camera matrices  $A_I$ ,  $A_J$  and  $A_K$  from the 20 tensor components of  $\mathbf{T}_{IJK}$ . Observe first that the camera matrices can never be recovered uniquely, since a multiplication by an arbitrary non-singular  $3 \times 3$  matrix to the right of  $T_{ijk}$  in (16) only changes the common scale of the tensor components. The following proposition maps  $\mathbf{T}_{IJK}$  to one set of compatible camera matrices.

**Proposition 3.** Given  $\mathbf{T}_{IJK}$  normalized such that  $\mathbf{t}^{111} = 1$ , the camera matrices can be calculated as

$$\begin{aligned} A_I &= \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{t}^{211} & KI\mathbf{e}^1 & -JI\mathbf{e}^1 \end{bmatrix} , \\ A_J &= \begin{bmatrix} 0 & 1 & 0 \\ -KJ\mathbf{e}^1 & \mathbf{t}^{121} & IJ\mathbf{e}^1 \end{bmatrix} \quad \text{and} \quad (21) \\ A_K &= \begin{bmatrix} 0 & 0 & 1 \\ JK\mathbf{e}^1 & -IK\mathbf{e}^1 & \mathbf{t}^{112} \end{bmatrix} . \end{aligned}$$

**Proof:** Since the camera matrices are only determined up to an affine transformation, the first rows of  $A_I$ ,  $A_J$  and  $A_K$  can be set to the  $3 \times 3$  identity. The remaining components are determined by straightforward calculations of the minors in (16).  $\square$

We now turn to the use of line correspondences to constrain the components of the affine tensors. According to (6) the direction of a line projects similar to the projection of a point except for the extra scale factor. Consider (6) for three different images of a line with direction  $\mathbf{D}$  in 3D space,

$$\begin{aligned} \lambda_I \mathbf{d}_I &= A_I \mathbf{D} , \\ \lambda_J \mathbf{d}_J &= A_J \mathbf{D} \quad \text{and} \\ \lambda_K \mathbf{d}_K &= A_K \mathbf{D} . \end{aligned}$$

Since these equations are linear in the scale factors and in  $\mathbf{D}$ , they can be written

$$N \begin{bmatrix} \mathbf{D} \\ -\lambda_I \\ -\lambda_J \\ -\lambda_K \end{bmatrix} = \begin{bmatrix} A_I & \mathbf{d}_I & 0 & 0 \\ A_J & 0 & \mathbf{d}_J & 0 \\ A_K & 0 & 0 & \mathbf{d}_K \end{bmatrix} \begin{bmatrix} \mathbf{D} \\ -\lambda_I \\ -\lambda_J \\ -\lambda_K \end{bmatrix} = 0 . \quad (22)$$

Thus the nullspace of  $N$  is non-empty, hence  $\det N = 0$ . Expanding this determinant, we get

$$\epsilon_{iiv'} \epsilon_{jj'} \epsilon_{kk'} \mathbf{t}^{ijk} \mathbf{d}_I^{i'} \mathbf{d}_J^{j'} \mathbf{d}_K^{k'} = 0 ,$$

i.e., a trilinear expression in  $\mathbf{d}_1$ ,  $\mathbf{d}_2$  and  $\mathbf{d}_3$  with coefficients that are the components of the centred affine tensor included in  $\mathbf{T}_{IJK}$ . Finally, we conclude that the direction of each line gives one constraint on the viewing geometry and that both points and lines can be used to constrain the tensor components<sup>2</sup>.

### 3.3. Reduced three-view constraints

It may seem superfluous to use 20 numbers to describe the viewing geometry of three affine cameras, since specializing the trifocal tensor (which has 27 components) for the projective camera, to the affine case, the number of components reduces to only 16 without using relative coordinates, cf. (Torr 1995). Since our 20 numbers describe all trilinear functions between three affine views, the comparison is not fair, even if the specialization of the trifocal tensor also encodes the information about the base points. It should be compared with the  $3 \times 16 = 48$  and  $3 \times 27 = 81$  components of all trifocal tensors between three affine views and three projective views, respectively. Although, it is possible to use a tensorial representation with only 12 components to describe the viewing geometry.

In order to obtain a smaller number of parameters, start again from (15) and  $\text{rank } M \leq 3$ . This time we will only consider the  $4 \times 4$  minors of  $M$  that contain both of the rows one and two, one of the rows three and four, and one of the rows five and six. There are in total 4 such minors and they are linear in the coordinates of  $\mathbf{x}_I$ ,  $\mathbf{x}_J$  and  $\mathbf{x}_K$ . Again, these trilinear expressions have coefficients that are minors of  $T_{IJK}$  in (16), but this time the only minors occurring are the ones containing either both rows from  $A_I$  and one from  $A_J$  or  $A_K$ , or one row from each one of  $A_I$ ,  $A_J$  and  $A_K$ .

**Definition 3.** The minors built up by rows  $i$ ,  $j$  and  $k$  from  $T_{IJK}$  in (16), where either  $i \in \{1, 2\}$ ,  $j \in \{3, 4\}$ ,  $k \in \{5, 6\}$  or  $i = 1$ ,  $j = 2$ ,  $k \in \{3, 4, 5, 6\}$ , will be called the reduced centred

affine tensors and the 12 components will be denoted by  $\mathbf{T}_{IJK}^r = (\mathbf{t}^{ijk}, {}_{JI}\mathbf{e}^j, {}_{KI}\mathbf{e}^k)$ , where  $\mathbf{e}$  denotes the previously defined centred affine epipoles and  $\mathbf{t}$  denotes the previously defined centred affine tensor in (17).

Observe that  $\mathbf{T}_{IJK}^r$  is built up by three different tensors, the tow centred affine epipoles,  ${}_{JI}\mathbf{e}^j$  and  ${}_{KI}\mathbf{e}^k$ , which are contravariant tensors and the third order tensor  $\mathbf{t}^{ijk}$ , which is contravariant in all indices.

Given the image coordinates in all three images, the chosen minors obtained from  $M$  give linear constraints on the 12 components of  $\mathbf{T}_{IJK}^r$ . There are in total 4 such linear constraints and they can be written

$$\epsilon_{iiv'} \mathbf{t}^{ijk} x_I^{i'} - {}_{KI}\mathbf{e}^k x_J^j + {}_{JI}\mathbf{e}^j x_K^k = 0 \quad (23)$$

for  $j = 1, 2$  and  $k = 1, 2$ , which can be written as

$$R^r t^r = 0 , \quad (24)$$

where  $R^r$  is a  $4 \times 12$  matrix containing relative image coordinates of the image point and  $t^r$  is a vector containing the 12 components of the reduced centred affine tensors. Observe again that the overall scale of the tensor components can not be determined. The number of linearly independent constraints for different number of point correspondences are given in the following proposition.

**Proposition 4.** Two corresponding points in 3 images give 4 linearly independent constraints on the reduced centred affine tensors. Three points give 8 linearly independent constraints and four or more points give 11 linearly independent constraints. Thus the tensor components can be linearly recovered from at least four point correspondences in 3 images.

**Proof:** See Appendix A.  $\square$

Again the camera matrices can be calculated from the 12 tensor components.

**Proposition 5.** Given  $\mathbf{T}_{IJK}^r$  normalized such that  $\mathbf{t}^{111} = 1$ , the camera matrices can be cal-

culated as

$$\begin{aligned} A_1 &= \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{t}^{211} & KI\mathbf{e}^1 & -JI\mathbf{e}^1 \end{bmatrix}, \\ A_2 &= \begin{bmatrix} 0 & 1 & 0 \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \quad \text{and} \\ A_3 &= \begin{bmatrix} 0 & 0 & 1 \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \end{aligned} \quad (25)$$

where

$$\begin{aligned} a_{21} &= (\mathbf{t}^{121} \mathbf{t}^{211} - \mathbf{t}^{221})/KI\mathbf{e}^1 \\ a_{22} &= \mathbf{t}^{121} \\ a_{23} &= (JI\mathbf{e}^2 - JI\mathbf{e}^1 \mathbf{t}^{121})/KI\mathbf{e}^1 \\ a_{31} &= (\mathbf{t}^{112} \mathbf{t}^{211} - \mathbf{t}^{212})/JI\mathbf{e}^1 \\ a_{32} &= (KI\mathbf{e}^2 - KI\mathbf{e}^1 \mathbf{t}^{112})/JI\mathbf{e}^1 \\ a_{33} &= \mathbf{t}^{112}. \end{aligned}$$

*Proof:* The form of the elements  $a_{22}$  and  $a_{33}$  follows by direct calculations of the determinants corresponding to  $\mathbf{t}^{212}$  and  $\mathbf{t}^{221}$ , respectively. The others follow from taking suitable minors and solving the linear equations.  $\square$

Using these combinations of tensors, a number of minimal cases appear for recovering the viewing geometry. In order to solve these minimal cases one has to take also the non-linear constraints on the tensor components into account. However, in the present work, we concentrate on developing a method to use points, lines and conics in a unified manner, when there is a sufficient number of corresponding features available to avoid the minimal cases.

#### 4. Factorization

Reconstruction using matching constraints is limited to a few views only. In this section, a factorization based technique is given that handle arbitrarily many views for corresponding points, lines and conics. The idea of factorization is simple, but still a robust and effective way of recovering structure and motion. Previously with the matching constraints only the centre of the conic was used, but there are obviously more constraints that could be used. After having described the

general factorization method, we show one possible way of incorporating this extra information.

Now consider  $m$  points or conics, and  $n$  lines in  $p$  images. (4) and (6) can be written as one single matrix equation (with relative coordinates),

$$\begin{aligned} S &= \begin{bmatrix} \mathbf{x}_{11} & \dots & \mathbf{x}_{1m} & \lambda_{11}\mathbf{d}_{11} & \dots & \lambda_{1n}\mathbf{d}_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{p1} & \dots & \mathbf{x}_{pm} & \lambda_{p1}\mathbf{d}_{p1} & \dots & \lambda_{pn}\mathbf{d}_{pn} \end{bmatrix} = \\ &= \begin{bmatrix} A_1 \\ \vdots \\ A_p \end{bmatrix} [\mathbf{X}_1 \dots \mathbf{X}_m \mathbf{D}_1 \dots \mathbf{D}_n]. \end{aligned} \quad (26)$$

The right-hand side of (26) is the product of a  $2p \times 3$  matrix and a  $3 \times (m+n)$  matrix, which gives the following theorem.

**Theorem 1.** The matrix  $S$  in (26) obeys

$$\text{rank } S \leq 3.$$

Observe that the matrix  $S$  contains entries obtained from measurements in the images, as well as the unknown scale factors  $\lambda_{ij}$ , which have to be estimated. The matrix is known as the measurement matrix. Assuming that these are known, the camera matrices, the 3D points and the 3D directions can be obtained by factorizing  $S$ . This can be done from the singular value decomposition of  $S$ ,  $S = U\Sigma V^T$ , where  $U$  and  $V$  are orthogonal matrices and  $\Sigma$  is a diagonal matrix containing the singular values,  $\sigma_i$ , of  $S$ . Let  $\tilde{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$  and let  $\tilde{U}$  and  $\tilde{V}$  denote the first three columns of  $U$  and  $V$ , respectively. Then

$$\begin{bmatrix} A_1 \\ \vdots \\ A_p \end{bmatrix} = \tilde{U} \sqrt{\tilde{\Sigma}} \quad \text{and} \quad (27)$$

$$[\mathbf{X}_1 \dots \mathbf{X}_m \mathbf{D}_1 \dots \mathbf{D}_n] = \sqrt{\tilde{\Sigma}} \tilde{V}^T$$

fulfil (26). Observe that the whole singular value decomposition of  $S$  is not needed. It is sufficient to calculate the three largest eigenvalues and the corresponding eigenvectors of  $SS^T$ . The only missing component is the scale factors  $\lambda_{ij}$  for the lines. These can be obtained in the following way.

Assume that  $\mathbf{T}_{IJK}$  or  $\mathbf{T}_{IJK}^r$  has been calculated. Then the camera matrices can be calculated

from Proposition 3 or Proposition 5. It follows from (22) that once the camera matrices for three images are known, the scale factors for each direction can be calculated up to an unknown scale factor. It remains to estimate the scale factors for all images with a consistent scale. We have chosen the following method. Consider the first three views with camera matrices  $A_1$ ,  $A_2$  and  $A_3$ . Rewriting (22) as

$$M \begin{bmatrix} \mathbf{D} \\ -1 \end{bmatrix} = \begin{bmatrix} A_1 & \lambda_1 \mathbf{d}_1 \\ A_2 & \lambda_2 \mathbf{d}_2 \\ A_3 & \lambda_3 \mathbf{d}_3 \end{bmatrix} \begin{bmatrix} \mathbf{D} \\ -1 \end{bmatrix} = 0, \quad (28)$$

shows that  $M$  in (28) has rank less than 4 which implies that all  $4 \times 4$  minors are equal to zero. These minors give linear constraints on the scale factors. However, only 3 of them are independent. So a system with the following appearance is obtained,

$$\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = 0, \quad (29)$$

where  $*$  indicates a matrix entry that can be calculated from  $A_i$  and  $\mathbf{d}_i$ . It is evident from (29) that the scale factors  $\lambda_i$  only can be calculated up to an unknown common scale factor. By considering another triplet, with two images in common with the first triplet, say the last two, we can obtain consistent scale factors for both triplets by solving a system with the following appearance,

$$\begin{bmatrix} * & * & * & 0 \\ * & * & * & 0 \\ * & * & * & 0 \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = 0.$$

In practice, all minors of  $M$  in (28) should be used. This procedure is easy to systematize such that all scale factors from the direction of one line can be computed as the nullspace of a single matrix. The drawback is of course that we first need to compute all camera matrices of the sequence. An alternative would be to reconstruct the 3D direction  $\mathbf{D}$  from one triplet of images according to (22) and then use this direction to solve for the scale factors in the other images.

In summary, the following algorithm is proposed.

1. Calculate the scale factors  $\lambda_{ij}$  using  $\mathbf{T}_{IJK}$  or  $\mathbf{T}_{IJK}^r$ .
2. Calculate  $S$  in (26) from  $\lambda_{ij}$  and the image measurements.
3. Calculate the singular value decomposition of  $S$ .
4. Estimate the camera matrices and the reconstruction of points and line directions according to (27).
5. Reconstruct 3D lines and 3D quadrics.

The last step needs a further comment. From the factorization, the 3D directions of the lines and the centres of the quadrics are obtained. The remaining unknowns can be recovered linearly from (5) for lines and (10) for quadrics.

Now to the question of how to incorporate all available constraints for the conics. Given that the quadrics in space are disk quadrics, the following modification of the above algorithm can be done. Consider a triplet of images, with known matching constraints. Choose a point on a conic curve in the first image, and then use the epipolar lines in the other two images to get the point-point correspondences on the other curves. In general, there is a two-fold ambiguity since an epipolar line intersects a conic at two points. The ambiguity is solved by examining the epipolar lines between the second and third image in the triplet. Repeating this procedure, point correspondences on the conic curves can be obtained throughout the sequence, and used in the factorization method as ordinary points.

## 5. Closure constraints

The drawback of all factorization methods is the difficulty in handling missing data, i.e., when all features are not visible in all images. In this section, an alternative method based on closure constraints, is presented that can handle missing data in a unified manner. Two related methods are also discussed.

Given the centred affine tensor and the centred affine epipoles, it is possible to calculate a representative for the three camera matrices. Since the reconstruction and the camera matrices are determined up to an unknown affine transformation,

only a representative can be calculated that differs from the true camera matrices by an affine transformation. When an image sequence with more than three images is treated, it is possible to first calculate a representative for the camera matrices  $A_1$ ,  $A_2$  and  $A_3$ , and a representative for  $A_2$ ,  $A_3$  and  $A_4$  and then merge these together. This is not a good solution since errors may propagate uncontrollably from one triplet to another. It would be better to use all available combinations of affine tensors and calculate all camera matrices at the same time. The solution to this problem is to use the closure constraints.

There are two different types of closure constraints in the affine case springing from the two-view and three-view constraints. To obtain the second order constraint, start by stacking camera matrices  $A_I$  and  $A_J$  like in (12), which results in a  $4 \times 3$  matrix. Duplicate one of the columns to obtain a  $4 \times 4$  matrix

$$B_{IJ} = \begin{bmatrix} A_I & A_I^n \\ A_J & A_J^n \end{bmatrix},$$

where  $A_I^n$  denotes the  $n$ :th column of  $A_I$ . Since  $B_{IJ}$  is a singular matrix (a repeated column), we have  $\det B_{IJ} = 0$ . Expanding  $\det B_{IJ}$  by the last column, for  $n = 1, 2, 3$ , gives

$$[IJ\mathbf{e}^2 \ -IJ\mathbf{e}^1] A_I + [JI\mathbf{e}^2 \ -JI\mathbf{e}^1] A_J = 0, \quad (30)$$

where  $IJ\mathbf{e}^1$  etc. denote the centred affine epipoles. Thus (30) gives one linear constraint on the camera matrices  $A_I$  and  $A_J$ .

To obtain the third order type of closure constraints consider the matrix  $T_{IJK}$  defined in (16) for the camera matrices  $A_I$ ,  $A_J$  and  $A_K$  and duplicate one of the columns to obtain a  $6 \times 4$  matrix

$$C_{IJK} = \begin{bmatrix} A_I & A_I^n \\ A_J & A_J^n \\ A_K & A_K^n \end{bmatrix},$$

where again  $A_I^n$  denotes the  $n$ :th column of  $A_I$ . Since  $C_{IJK}$  has a repeated column it is rank deficient, i.e.,  $\text{rank } C_{IJK} < 4$ . Expanding the  $4 \times 4$  minors of  $C_{IJK}$  give three expressions, involving only two cameras of the same type as (30) and 12 expressions involving all three cameras of the type

$$\begin{aligned} [\mathbf{t}^{211} - \mathbf{t}^{111}] A_I + [KI\mathbf{e}^1 \ 0] A_J + \\ + [JI\mathbf{e}^1 \ 0] A_K = 0. \end{aligned} \quad (31)$$

Thus we get in total 15 linear constraints on the camera matrices  $A_I$ ,  $A_J$  and  $A_K$ . However, there are only 3 linearly independent constraints among these 15, which can easily be checked by using a computer algebra package, such as MAPLE. Some of these constraints involve only components of the reduced affine tensors, e.g., the one in (31), making it possible to use the closure constraints in the reduced case also.

To sum up, every second order combination of centred affine epipoles gives one linear constraint on the camera matrices and every third order combination of affine tensors gives 12 additional linear constraints on the camera matrices. Using all available combinations, all the linear constraints on the camera matrices can be stacked together in a matrix  $M$ ,

$$MA = M \begin{bmatrix} A_1 \\ \vdots \\ A_m \end{bmatrix} = 0. \quad (32)$$

Given a sufficient number of constraints on the camera matrices, they can be calculated linearly from (32). Observe that the nullspace of  $M$  has dimension 2, which implies that only the linear space spanned by the columns of  $A$  can be determined. This means that the camera matrices can only be determined up to an unknown affine transformation.

When only the second order combinations are used, it is not sufficient to use only the combinations between every successive pair of images. However, it is sufficient to use the combinations between views  $i, i+1$  and  $i, i+2$  for every  $i$ . This can easily be seen from the fact that one new image gives two new independent variables in the linear system of equations in (32) and the two new linear constraints balances this. When the third order combinations are used, it is sufficient to use the tensor combinations between views  $i, i+1, i+2$  for every  $i$ , which again can be seen by counting the number of unknowns and the number of linearly independent constraints. This is also the case for the reduced third order combinations.

The closure constraints bring the camera matrices,  $A_i$ ,  $i = 1, \dots, m$ , into the same affine coordinate system. However, the last column in the camera matrices, denoted by  $b_i$ , cf. (3), needs also to be calculated. These columns depend on the

chosen centroid for the relative coordinates. But if the visible feature configuration changes, as there may be missing data, the centroid changes as well. This has to be considered. For example, let  $\mathbf{x}_{0_1}$ ,  $\mathbf{x}_{0_2}$ ,  $\mathbf{x}_{0_3}$  and  $\mathbf{X}_0$  denote the centroid of the visible points in the images and in space for the first three views, respectively, and let  $\mathbf{x}_{0'_2}$ ,  $\mathbf{x}_{0'_3}$ ,  $\mathbf{x}_{0'_4}$  and  $\mathbf{X}_0'$  denote the centroid in the images and in space for views two, three and four, respectively. The centroids are projected as

$$\begin{aligned}\mathbf{x}_{0_i} &= A_i \mathbf{X}_0 + b_i, \quad i = 1, 2, 3 \quad \text{and} \\ \mathbf{x}_{0'_j} &= A_j \mathbf{X}_0' + b_j, \quad j = 2, 3, 4 \quad .\end{aligned}$$

This is a linear system in the unknowns  $b_1, b_2, b_3, b_4, \mathbf{X}_0$  and  $\mathbf{X}_0'$ . It is straightforward to generalize the above equations for  $m$  consecutive images and the system can be solved by a single SVD.

### 5.1. Related work

We examine two closely related algorithms for dealing with missing data.

Tomasi and Kanade propose one method in (Tomasi and Kanade 1992) to deal with the missing data problem for point features. In their method, one first locates a rectangular subset of the measurement matrix  $S$  (26) with no missing elements. Factorization is applied to this matrix. Then, the initial sub-block is extended row-wise (or column-wise) by propagating the partial structure and motion solution. In this way, the missing elements are filled in iteratively. The result is finally refined using steepest descent minimization.

As pointed out by Jacobs (Jacobs 1997), their solution seems like a reasonable heuristics, but the method has several potential disadvantages. First, the problem of finding the largest full submatrix of a matrix is NP-hard, so heuristics must be used. Second, the data is not used in a unified manner. As only a small subset is used in the first factorization, the initial structure and motion may contain significant inaccuracies. In turn, these errors may propagate uncontrollably as additional rows (or columns) are computed. Finally, the refinements with steepest descent is not guaranteed to converge to the globally optimal solution.

The method proposed in (Jacobs 1997) also starts with the measurement matrix  $S$  using only

points. Since  $S$  should be of rank three, the  $m$  columns of  $S$  span a 3-dimensional linear subspace, denoted  $\mathcal{L}$ . Consequently, the span of any three columns of  $S$  should intersect the subspace  $\mathcal{L}$ . If there are missing elements in any of the three columns, the span of the triplet will be of higher dimension. In that case, the constraint that the subspace  $\mathcal{L}$  should lie in the span of the triplet will be a weaker one. In practise, Jacobs calculates the nullspace of randomly chosen triplets, and finally, the solution is found by computing the nullspace of the span of the previously calculated nullspaces, using SVD.

Jacobs' method is closely related to the closure constraints. It can be seen as the 'dual' of the closure constraints, since it generates constraints by picking columns in the measurement matrix, while we generate constraints by using rows. Therefore, a comparison based on numerical experiments has been performed, which is presented in the experimental section.

There are also significant differences. First, by using matching tensors, lines can also contribute to constraining the viewing geometry. Second, for  $m$  points, there are  $\binom{m}{3}$  point triplets. In practice, this is hard to deal with, so Jacobs heuristically chooses a random subset of the triplets, without knowing if it is sufficient. With our method we know that, e.g., it is sufficient to use every consecutive third order closure constraint. Finally, Jacobs uses the visible point configuration in adjacent images to calculate the centroid. Since there is missing data, this approximation often leads to significant errors (see experimental comparison). However, one may modify Jacobs' method, so it correctly compensates for the centroid. In order to make a fair experimental comparison, we have included a modified version which properly handles this problem. It works in the same manner as the original one, but it does not use relative coordinates. In turn, it has to compute a 4-dimensional linear subspace of the measurement matrix  $S$ . This modified version generates constraints by picking quadruples of columns in  $S$ . Since there are  $\binom{m}{4}$  quadruples, the complexity is much worse than the original one.

## 6. Experiments

The presented methods have been tested and evaluated on both synthetic and real data.

### 6.1. Simulated data

All synthetic data was produced in the following way. First, points, line segments and conics were randomly distributed in space with coordinates between  $-500$  and  $+500$  units. The camera positions were chosen at a nominal distance around 1000 units from the origin and then all 3D features were projected to these views and the obtained images were around  $500 \times 500$  pixels. In order to test the stability of the proposed methods, different levels of noise were added to the data. Points were perturbed with uniform, independent Gaussian noise. In order to incorporate the higher accuracy of the line segments, a number of evenly sampled points on the line segments were perturbed with independent Gaussian noise in the normal direction of the line. Then, the line parameters were estimated with least-squares. The conics were handled similarly. The residual error for points was chosen as the distance between the true point position and the re-projected reconstructed 3D point. For lines, the residual errors were chosen as the smallest distances between the endpoints of the true line segment and the re-projected 3D line. For conics, the errors were measured with respect to the centroid. These settings are close to real life situations. All experiments were repeated 100 times and the results reflect the average values. Before the actual computations, all input data was rescaled to improve numerical conditioning.

In Table 1, it can be seen that the 20-parameter formulation (the centred affine tensor and the centred affine epipoles) of three views is in general superior to the 12-parameter formulation (the reduced affine tensors). For three views, factorization gives slightly better results. All three methods handle moderate noise perturbations well. In Table 2 the number of points and lines is varied. In general, the more number of points and lines the better results, and the non-reduced representation is still superior the reduced version. Finally, in Table 3 the number of views is varied. In this experiment, two variants of the fac-

torization method are tried and compared to the method of closure constraints. The first one (I) uses the centroid of the conic as a point feature, and the second one uses, in addition, one point on each conic curve, obtained by the epipolar transfer (see Section 4). The first method appears more robust than the second one, even though the second method incorporates all the constraints of the conic. Somewhat surprisingly, the method based on closure constraints has similar performance as the best factorization method<sup>3</sup>. The closure constraints are of third order and only the tensors between views  $i$ ,  $i + 1$  and  $i + 2$  have been used. However, the differences are minor between the two methods and they both manage to keep the residuals low.

Table 1. Result of simulations of 10 points and 10 lines in 3 images for different levels of noise using the third order combination of affine tensors, the reduced third order combination of affine tensors and the factorization approach. The root mean square (RMS) errors are shown for the reduced affine tensors  $\mathbf{T}_{IJK}^r$ , the non-reduced  $\mathbf{T}_{IJK}$ , and factorization.

STD of noise	0	1	2	5
<b>Red. affine tensors</b>				
RMS of points	0.0	3.3	8.4	7.7
RMS of lines	0.0	3.5	7.1	8.6
<b>Affine tensors</b>				
RMS of points	0.0	1.6	2.2	6.2
RMS of lines	0.0	1.7	2.3	8.3
<b>Factorization</b>				
RMS of points	0.0	1.0	1.8	4.5
RMS of lines	0.0	1.1	2.1	6.8

Table 2. Results of simulation of 3 views with a different number of points and lines and with a standard deviation of noise equal to 1. The table shows the resulting error (RMS) after using the reduced affine tensors  $\mathbf{T}_{IJK}^r$ , the non-reduced  $\mathbf{T}_{IJK}$ , and factorization.

#points, #lines	3,3	5,5	10,10	20,20
<b>Red. affine tensors</b>				
RMS of points	1.0	1.5	1.6	2.0
RMS of lines	3.9	1.5	1.2	1.7
<b>Affine tensors</b>				
RMS of points	1.0	1.6	1.0	1.2
RMS of lines	3.9	2.2	0.8	1.1
<b>Factorization</b>				
RMS of points	1.0	1.1	0.9	0.9
RMS of lines	3.9	1.1	0.7	0.7

## 6.2. Real data

Two sets of images have been used in order evaluate the different methods. The first set is used to verify the performance on real images, and the second set is used for a comparison with the method of Jacobs.

**6.2.1. Statue sequence** A sequence of 12 images was taken of an outdoor statue containing both points, lines and conics. More precisely, the statue consists of two ellipses lying on two different planes in space and the two ellipses are connected by straight lines, almost like a hyperboloid, see Figure 1. There are in total 80 lines between the ellipses. In total, four different experiments were performed on these images.

In the first three experiments only 5 images were used. In these images, 17 points, 17 lines and the 2 ellipses were picked out by hand in all images. For the ellipses and the lines, the appropriate representations were calculated by least-squares.

In the first experiment, only the second order closure constraints between images  $i$  and  $i + 1$  and between images  $i$  and  $i + 2$  were used. The reconstructed points, lines and conics were obtained by intersection using the computed camera matrices. The detected and re-projected features are shown in Figure 1.

In the second experiment, only the third order closure constraints between images  $i$ ,  $i + 1$  and  $i + 2$  were used. The tensors were estimated from

both point, line and conic correspondences. The camera matrices were calculated from the closure constraints and the 3D features were obtained by intersection. The detected and re-projected features are shown in Figure 2 together with the reconstructed 3D model.

The third experiment was performed on the same data as the first two, but the factorization method was applied. In Figure 2, a comparison is given for the three methods. The third order closure constraints yield better results than the second order constraints as expected. However, the factorization method is outperformed by the third order closure constraints which was unexpected.

Table 3. Table showing simulated results for 10 points, 10 lines and 3 conics in a different number of views, with an added error of standard deviation 1 for the factorization approaches and using third order closure constraints. Factorization I uses only conic centres, while Factorization II uses an additional point on each conic curve.

#views	3	5	10	20
Factorization I				
RMS of points	0.84	0.73	0.69	0.65
RMS of lines	0.62	0.62	0.70	0.73
RMS of conics	1.00	0.76	0.78	0.76
Factorization II				
RMS of points	0.87	1.00	1.25	1.59
RMS of lines	0.67	0.98	1.45	1.90
RMS of conics	1.02	1.07	1.43	1.71
Closure Constr.				
RMS of points	0.86	0.75	0.68	0.65
RMS of lines	0.64	0.64	0.70	0.75
RMS of conics	1.20	0.84	0.86	0.85

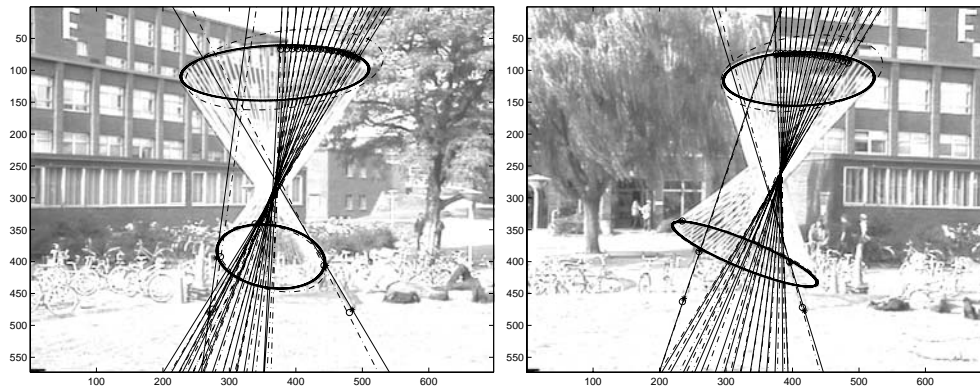


Fig. 1. The second and fourth image of the sequence, with detected points lines and conics together with re-projected points, lines and conics using the second order closure constraints.

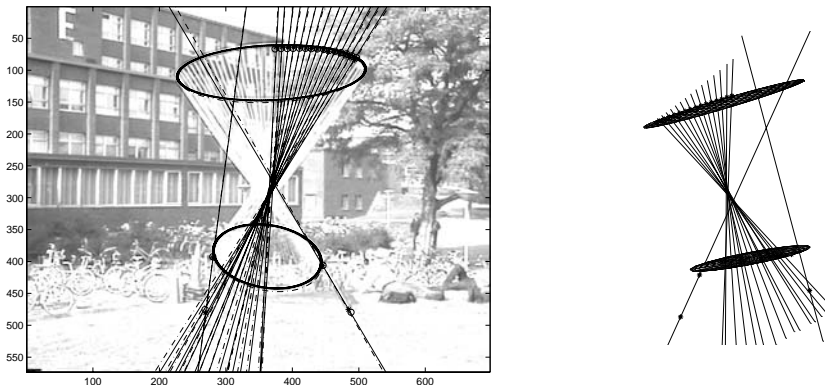


Fig. 2. The second image of the sequence, with detected and re-projected points, lines and conics together with the reconstructed 3D model using the third order closure constraints.

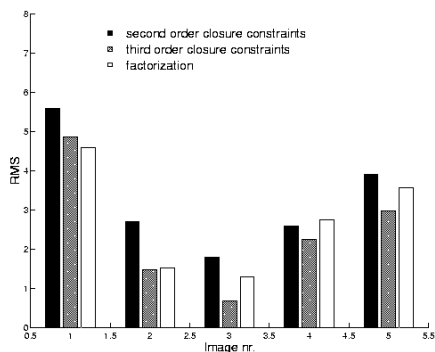


Fig. 3. Root mean square (RMS) error of second and third order closure constraints, and factorization for five images in the statue sequence.

The final experiment was performed on all 12 images of the statue. In these images, there is a lot of missing data, i.e., all features are not visible in all images. The reconstruction then has to be based on the closure constraints. In the resulting 3D model, the two ellipses and the 80 lines were reconstructed together with 80 points, see Figure 4. The resulted structure and motion was also refined using bundle adjustment techniques, cf. (Atkinson 1996) to get, in a sense, an optimal reconstruction, and compared to that of the original one. To get an idea of the errors caused by the affine camera model, the result was also used as initialization for a bundle adjustment algorithm based on the projective camera model. The com-

parison is given in Figure 5, image per image. The quality of the output from the method based on the closure constraints is not optimal, but fairly accurate. If further accuracy is required, it can serve as a good initialization to a bundle adjustment algorithm for the affine or the full projective/perspective model.

**6.2.2. Box sequence** As a final test, we have compared our method to that of Jacobs, described in (Jacobs 1997). Naturally, we can only use point features, since Jacobs method is only valid for that. As described in Section 5.1, it works by finding a rank three approximation of the measurement matrix. Since this original version incorrectly compensates for the translational component, we have included a modified version which does this properly by finding a rank four approximation. We have used Jacobs' own implementation in Matlab, for both versions.

As a test sequence, we have chosen the box sequence, which was also used by Jacobs in his paper. The sequence, which originates from the Computer Vision Laboratory at the University of Massachusetts, contains forty points tracked across eight images. One frame is shown in Figure 6. We generated artificial occlusions, by assuming that each point is occluded for some fraction of the sequence. The fraction is randomly chosen for each point from a uniform distribution. These settings are the same as in (Jacobs 1997). For Jacobs' algorithm, the maximum number of triplets (quadruples) has been set to the actual

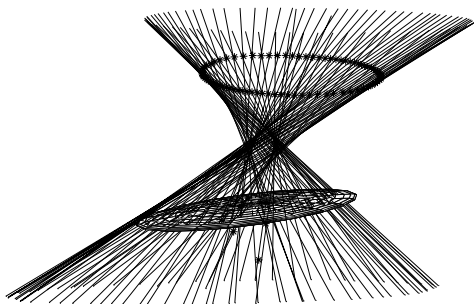


Fig. 4. The full reconstruction of the statue based on the third order closure constraints.

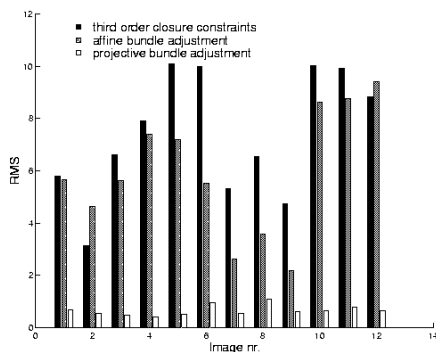


Fig. 5. Root mean square (RMS) error of third order closure constraints, affine bundle adjustment and projective bundle adjustment for each image in the statue sequence.

number of available triplets (quadruples). However, this is only an upper limit. Jacobs chooses triplets until the nullspace matrix of all triplets occupies ten times as many columns as the original measurement matrix. We have set this threshold to 100 times. In turn, all possible third order closure constraints for the sequence are calculated.

In Figure 7, the result is graphed for Jacobs' rank three approximation, rank four approximation and the method of closure constraints. The result for the rank three version is clearly biased. The performance of the rank four and closure based methods are similar up to about 30 percent missing data. With more missing data, the closure method is superior. Based on this exper-

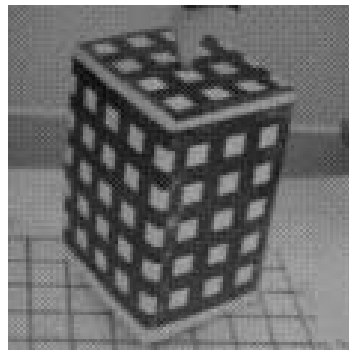


Fig. 6. One image of the box sequence.

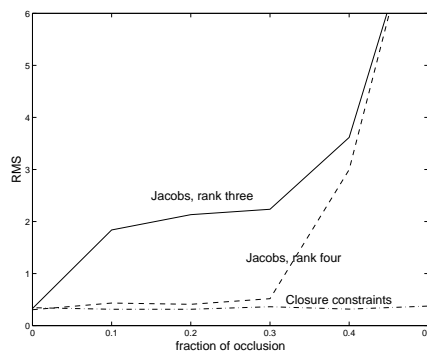


Fig. 7. Averaged RMS error over 100 trials. The error is plotted against the average fraction of frames in which a point is occluded. The tested methods are Jacobs' rank three and four methods and closure based method.

iment, the closure constraints are preferable both in terms of stability and complexity.

## 7. Conclusions

In this paper, we have presented an integrated approach to the structure and motion problem for the affine camera model. Correspondences of points, lines and conics have been handled in a unified manner to reconstruct the scene and the camera positions. The proposed scheme is illustrated on both simulated and real data.

## Appendix A

Proof: (of Proposition 2) The number of linearly independent equations (in the components of  $\mathbf{T}_{IJK}$ ) can be calculated as follows. The 15 linear constraints obtained from the minors of  $M$  in (15) are not linearly independent, i.e., there exists non-trivial combinations of these constraints that vanishes. Consider the matrix

$$\begin{bmatrix} A_I & \mathbf{x}_I & \mathbf{x}_I \\ A_J & \mathbf{x}_J & \mathbf{x}_J \\ A_K & \mathbf{x}_K & \mathbf{x}_K \end{bmatrix}$$

obtained from  $M$  by duplicating its last column. This matrix is obviously of rank  $< 5$ , implying that all  $5 \times 5$  minors vanish. There are 6 such minors and they can be written (using Laplacian expansions) as linear equations in the previously obtained linear constraints (minors from the first four columns) with image coordinates (elements from the last column) as coefficients. This gives 6 linear dependencies on the 15 original constraints, called second order constraints. On the other hand it is obvious that all linear constraints on the originally obtained 15 constraints can be written as the vanishing of minors from a determinant of the form

$$\begin{bmatrix} A_I & \mathbf{x}_I & k_1 \\ A_J & \mathbf{x}_J & k_2 \\ A_K & \mathbf{x}_K & k_3 \end{bmatrix}.$$

Hence the vector  $[k_1 k_2 k_3]^T$  is a linear combination of the other columns of the matrix and since it has to be independent of  $A_I$ ,  $A_J$  and  $A_K$ , we deduce that we have obtained all possible second order linear constraints.

The process does not stop here, since these second order constraints are not linearly independent. This can be seen by considering the matrix

$$\begin{bmatrix} A_I & \mathbf{x}_I & \mathbf{x}_I & \mathbf{x}_I \\ A_J & \mathbf{x}_J & \mathbf{x}_J & \mathbf{x}_J \\ A_K & \mathbf{x}_K & \mathbf{x}_K & \mathbf{x}_K \end{bmatrix}.$$

Again Laplacian expansions give one third order constraint. To sum up we have  $15 - (6 - 1) = 10$  linearly independent constraints for two corresponding points. The similar reasoning as before gives that all possible second order constraints has been obtained.

Using three corresponding points we obtain 10 linearly independent constraints from the second point and 10 linearly independent constraints from the third point. However, there are linear dependencies among these 20 constraints. To see this consider the matrix

$$\begin{bmatrix} A_I & \mathbf{x}_I & \bar{\mathbf{x}}_I \\ A_J & \mathbf{x}_J & \bar{\mathbf{x}}_J \\ A_K & \mathbf{x}_K & \bar{\mathbf{x}}_K \end{bmatrix},$$

where  $\bar{\mathbf{x}}$  denotes the third point. Using Laplacian expansions of the  $5 \times 5$  minors we obtain 6 bilinear expressions in  $\mathbf{x}$  and  $\bar{\mathbf{x}}$  with the components of the third order combination of affine tensors as coefficients. Each such minor give a linear dependency between the constraints, i.e., 6 second order constraints. Again there are third order constraints obtained from

$$\begin{bmatrix} A_I & \mathbf{x}_I & \mathbf{x}_I & \bar{\mathbf{x}}_I \\ A_J & \mathbf{x}_J & \mathbf{x}_J & \bar{\mathbf{x}}_J \\ A_K & \mathbf{x}_K & \mathbf{x}_K & \bar{\mathbf{x}}_K \end{bmatrix}$$

and

$$\begin{bmatrix} A_I & \mathbf{x}_I & \bar{\mathbf{x}}_I & \bar{\mathbf{x}}_I \\ A_J & \mathbf{x}_J & \bar{\mathbf{x}}_J & \bar{\mathbf{x}}_J \\ A_K & \mathbf{x}_K & \bar{\mathbf{x}}_K & \bar{\mathbf{x}}_K \end{bmatrix},$$

giving in total 2 third order constraints. To sum up we have  $2 \times 10 - (6 - 1 - 1) = 16$  independent constraints. We note again that all possible linear constraints have been obtained according to the same reasoning as above.

The same analysis can be made for the case of four point matches. First we have 10 linearly independent constraints from each point (apart from the first one) and each pair of corresponding points give 4 second order linear constraints, giving  $3 \times 10 - 3 \times 4 = 18$  constraints. Then one third order constraint can be obtained from the determinant of

$$\begin{bmatrix} A_I & \mathbf{x}_I & \bar{\mathbf{x}}_I & \hat{\mathbf{x}}_I \\ A_J & \mathbf{x}_J & \bar{\mathbf{x}}_J & \hat{\mathbf{x}}_J \\ A_K & \mathbf{x}_K & \bar{\mathbf{x}}_K & \hat{\mathbf{x}}_K \end{bmatrix},$$

where  $\hat{\mathbf{x}}$  denote the fourth point, giving  $18 - (-1) = 19$  linearly independent constraints for four points. Again all possible constraints have been obtained, which concludes the proof.  $\square$

Remark. The rank condition  $\text{rank } M < 4$  is equivalent to the vanishing of all  $4 \times 4$  minors of  $M$ . These minors are algebraic equations in the

24 elements of  $M$ . These (non-linear) equations define a variety in 24 dimensional space. The dimension of this variety is a well-defined number, in this case 21, which means that the co-dimension is 3. This means that, in general (at all points on the variety except for a subset of measure zero in the Zariski topology), the variety can locally be described as the vanishing of three polynomial equations. This can be seen by making row and column operations on  $M$  until it has the following structure

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & p \\ 0 & 0 & 0 & q \\ 0 & 0 & 0 & r \end{bmatrix},$$

where  $p$ ,  $q$  and  $r$  are polynomial expressions in the entries of  $M$ . The matrix above has rank  $< 4$  if and only if  $p = q = r = 0$ , i.e., three polynomial equations define the variety locally. The points on the variety where the rank condition can not locally be described by three algebraic equations are the ones where all of the  $3 \times 3$  minors of  $M$  vanishes, which is a closed (and hence of measure zero) subset in the Zariski topology.

**Remark.** Since we are interested in linear constraints, we obtain 10 linearly independent equations instead of the 3 so-called algebraically independent equations described in the previous remark. However, one can not select 10 such constraints in advance that will be linearly independent for every point match. Therefore, in numerical computations, it is better to use all of them.

**Proof:** (of Proposition 4) It is easy to see that there are no second (or higher) order linear constraints involving only the 4 constraints in (23). Neither are there any higher order constraints for the two sets of (23) involving two different points,  $\mathbf{x}$  and  $\bar{\mathbf{x}}$ . Finally, for four different points, there can be no more than 11 linearly independent constraints, since according to (24) the matrix containing all constraints has a non-trivial null-space.  $\square$

## Notes

1. Again the choice of defining a contravariant tensor is arbitrarily made. In fact, the tensor could have been defined covariantly as

$$\mathbf{t}_{ijk} = \epsilon_{ii'} \epsilon_{jj'} \epsilon_{kk'} \det \begin{bmatrix} A_{J'}^{i'} \\ A_{J'}^j \\ A_{K'}^{k'} \end{bmatrix}$$

which is the one used in (Quan and Kanade 1997). Transformations between these representations (and other intermediate ones such as covariant in one index and contravariant in the other ones) can easily be made.

2. The tensor  $\mathbf{t}_{ijk}$  can also be used to transfer directions seen in two of the three images to a direction in the third one, using the mixed form  $\mathbf{t}_{jk}^i$  according to

$$\mathbf{d}_I^i = \mathbf{t}_{jk}^i \mathbf{d}_J^j \mathbf{d}_K^k .$$

3. This has been confirmed under various imaging conditions, like e.g., closely spaced images.

## References

- Atkinson, K. B.: 1996, *Close Range Photogrammetry and Machine Vision*, Whittles Publishing.
- Bretzner, L. and Lindeberg, T.: 1998, Use your hand as a 3-d mouse, or, relative orientation from extended sequences of sparse point and line correspondences using the affine trifocal tensor, Proc. 5th European Conf. on Computer Vision, Freiburg, Germany.
- Faugeras, O. D.: 1992, What can be seen in three dimensions with an uncalibrated stereo rig?, in G. Sandini (ed.), Proc. 2nd European Conf. on Computer Vision, Santa Margherita Ligure, Italy, Springer-Verlag, pp. 563–578.
- Heyden, A.: 1995, *Geometry and Algebra of Multiple Projective Transformations*, PhD thesis, Lund Institute of Technology, Sweden.
- Jacobs, D.: 1997, Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images, Proc. Conf. Computer Vision and Pattern Recognition, pp. 206–212.
- Kahl, F. and Heyden, A.: 1998, Structure and motion from points, lines and conics with affine cameras, Proc. 5th European Conf. on Computer Vision, Freiburg, Germany.
- Koenderink, J. J. and van Doorn, A. J.: 1991, Affine structure from motion, J. Opt. Soc. America 8(2), 377–385.
- Ma, S.: 1993, Conic-based stereo, motion estimation, and pose determination, Int. Journal of Computer Vision 10(1), 7–25.
- Maybank, S.: 1993, *Theory of Reconstruction from Image Motion*, Springer-Verlag, Berlin, Heidelberg, New York.
- McLauchlan, P. F. and Murray, D. W.: 1995, A unifying framework for structure and motion recovery from image sequences, Proc. 5th Int. Conf. on Computer

- Vision, MIT, Boston, MA, IEEE Computer Society Press, Los Alamitos, California, pp. 314–320.
- Mundy, J. L. and Zisserman, A. (eds): 1992, Geometric invariance in Computer Vision, MIT Press, Cambridge Ma, USA.
- Quan, L. and Kanade, T.: 1997, Affine structure from line correspondences with uncalibrated affine cameras, IEEE Trans. Pattern Analysis and Machine Intelligence 19(8).
- Quan, L. and Ohta, Y.: 1998, A new linear method for euclidean motion/structure from three calibrated affine views, Proc. Conf. Computer Vision and Pattern Recognition, Santa Barbara, USA, pp. 172–177.
- Semple, J. G. and Kneebone, G. T.: 1952, Algebraic Projective Geometry, Clarendon Press, Oxford.
- Shapiro, L. S.: 1995, Affine Analysis of Image Sequences, Cambridge University Press.
- Shapiro, L. S., Zisserman, A. and Brady, M.: 1995, 3d motion recovery via affine epipolar geometry, Int. Journal of Computer Vision 16(2), 147–182.
- Shashua, A. and Navab, N.: 1996, Relative affine structure: Canonical model for 3d from 2d geometry and applications, IEEE Trans. Pattern Anal. Machine Intell 18(9), 873–883.
- Sparr, G.: 1996, Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences, Proc. Int. Conf. on Pattern Recognition, Vienna, Austria.
- Sturm, P. and Triggs, B.: 1996, A factorization based algorithm for multi-image projective structure and motion, Proc. 4th European Conf. on Computer Vision, Cambridge, UK, pp. 709–720.
- Tomasi, C. and Kanade, T.: 1992, Shape and motion from image streams under orthography: a factorization method, Int. Journal of Computer Vision 9(2), 137–154.
- Torr, P.: 1995, Motion Segmentation and Outlier Detection, PhD thesis, Department of Engineering Science, University of Oxford.
- Triggs, B.: 1996, Factorization methods for projective structure and motion, Proc. Conf. Computer Vision and Pattern Recognition.
- Triggs, B.: 1997, Linear projective reconstruction from matching tensors, Image and Vision Computing 15(8), 617–625.
- Weng, J., Huang, T. and Ahuja, N.: 1992, Motion and structure from line correspondences: Closed-form solution, uniqueness, and optimization, IEEE Trans. Pattern Analysis and Machine Intelligence 14(3), 318–336.