Structure and Motion from Points, Lines and Conics with Affine Cameras

Fredrik Kahl*, Anders Heyden**

Dept of Mathematics, Lund University
Box 118, S-221 00 Lund, Sweden
email: {fredrik,heyden}@maths.lth.se

Abstract. *In this paper we present an integrated approach that solves the structure and motion problem for affine cameras. Given images of corresponding points, lines and conics in any number of views, a reconstruction of the scene structure and the camera motion is calculated, up to an affine transformation. Starting with three views, two novel concepts are introduced. The first one is a quasi-tensor consisting of* 20 *components and the second one is another quasi-tensor consisting of* 12 *components. These tensors describe the viewing geometry for three views taken by an affine camera. It is shown how correspondences of points, lines and conics can be used to constrain the tensor components. A set of affine camera matrices compatible with the quasi-tensors can easily be calculated from the tensor components. The resulting camera matrices serve as an initial guess in a factorisation method, using points, lines and conics concurrently, generalizing the well-known factorisation method by Tomasi-Kanade. Finally, examples are given that illustrate the developed methods on both simulated and real data.*

1 Introduction

One of the main problems in computer vision is to recover the scene structure and the camera motion from a set of images. In the last few years there has been an intense research on this subject, especially concentrated on point features. Recently, attention has also turned to the use of other features such as lines, conics, general curves or even silhouettes of surfaces, see [11, 10, 1, 7].

For points, the viewing geometry can be estimated linearly in two, three and four images, see [2, 5, 19]. For more images, a major breakthrough was made in [17] where a factorisation method was developed in the case of an orthographic camera. This has later been generalized to the projective camera, cf. [16, 15, 6].

The next natural step is to use line correspondences. Linear algorithms exist for the case of three images obtained by a projective camera, see [20, 4]. In the latter algorithm, it is possible to combine points and lines in order to estimate

* Supported by the ESPRIT Reactive LTR project 21914, CUMULI
** Supported by the Swedish Research Council for Engineering Sciences (TFR), project 95-64-222

the viewing geometry. With the projective camera at least 13 line correspondences are needed to estimate the trifocal tensor linearly. However, with the affine camera it is possible to recover the viewing geometry from a minimum of seven line correspondences in three views, cf. [12].

In the case of conics, the situation is more complicated. In [11] a conic in space was reconstructed from two views with known epipolar geometry. A further step was taken in [8], where conic correspondences in two images were used to estimate the epipolar geometry, but the methods rely on non-linear algorithms that require initialisations.

In this paper we present an integrated approach for structure and motion from corresponding points, lines and conics. From any number of views with any number of image features, we show how to recover the scene structure and the camera motion with an affine camera. First, we derive and analyse two different parametrisations describing the viewing geometry for three cameras. The components of these parametrisations are combinations of different tensors and therefore called quasi-tensors. Specializing the projective trifocal tensor directly to the affine case leads to a larger number of parameters, cf. [18]. The corresponding image features impose constraints on the coefficients of the quasi-tensor, making it possible to estimate it when a sufficient number of correspondences is available. When a quasi-tensor is known the camera matrices for these three views are easily determined up to an unknown affine transformation. Then, a factorisation algorithm is presented that uses all points, lines and conics in all images concurrently to estimate structure and motion. The reconstruction can optionally be refined with bundle adjustment.

The motivation for using affine cameras instead of projective ones is manyfold. Firstly, in most practical applications, the affine camera model is a good approximation to the projective one. In addition, an affine reconstruction of the scene is obtained, whilst in the projective case only projective structure is recovered. Secondly, algorithms using the full projective model are inherently unstable in situations where the depth of the scene is small compared to the viewing distance. In such situations, it is even advisable to use the affine model to get more robust results. Thirdly, there is a lack of satisfactory algorithms for non-point features for projective cameras. Another advantage is that the minimum number of corresponding image features required is substantially smaller.

2 The Affine Camera

In this section we give a brief review of the affine camera model and describe how points and lines are projected onto the image. For a more thorough treatment, see [14] for points and [12] for lines. Then, we analyse how quadrics in the scene are projected to conics in the image.

The projective/perspective camera is modeled by

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

where P denotes the standard 3×4 camera matrix and λ a scale factor. Here $\mathbf{X} = [X\,Y\,Z]^T$ and $\mathbf{x} = [x\,y]^T$ denote point coordinates in the 3D scene and in the image respectively.

The affine camera model, first introduced by Mundy and Zisserman in [9], has the same form as in (1), but the camera matrix is restricted to

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

and the homogeneous scale factors λ are 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. The affine camera has eight degrees of freedom, since (2) is only defined up to a scalar factor, and it can be seen as a projective camera with its optical center on the plane at infinity.

Rewriting (1) with the affine camera matrix (2) results in

$$\mathbf{x} = A\mathbf{X} + b \ , \tag{3}$$

where

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

By using relative coordinates with respect to some reference point $\mathbf{X_0}$ in the object and to the point $\mathbf{x_0} = A\mathbf{X_0} + b$ in the image, (3) simplifies to

$$\Delta\mathbf{x} = A\Delta\mathbf{X} \ , \tag{4}$$

where $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x_0}$ and $\Delta\mathbf{X} = \mathbf{X} - \mathbf{X_0}$. Normally, the reference point is chosen as the centroid of the set of points. This is possible since the centroid of the 3D points projects onto the centroid of the image points.

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 $\mathbf{l}$ according to

$$\mathbf{l} = A\mathbf{L} + b = A(\mathbf{X} + \mu\mathbf{D}) + b = A\mathbf{X} + \mu A\mathbf{D} + b = \mathbf{x} + \mu A\mathbf{D} \ , \tag{5}$$

where $\mathbf{x} = A\mathbf{X} + b$. From (5), it follows that the direction $\mathbf{d}$ of the image line is obtained as

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

In [12], it was noted that this equation is nothing but a projective transformation from $\mathbb{P}^2$ to $\mathbb{P}^1$ if the directions are regarded as points in $\mathbb{P}^2$ and $\mathbb{P}^1$, respectively. 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 λ present in (6), but not in (4). Thus, with known scale factor λ, a direction can be treated as an ordinary point. This fact will be used later on in the factorisation algorithm.

A general *conic curve* in the plane can be represented by its dual form, the *conic envelope*,

$$\mathbf{u}^T l \mathbf{u} = 0 \ , \tag{7}$$

where l denotes a 3×3 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*,

$$\mathbf{U}^T L \mathbf{U} = 0 \ , \tag{8}$$

where L denotes a 4×4 symmetric matrix and $\mathbf{U} = [U\, V\, W\, 1]^T$ denotes extended dual coordinates in the 3D space. For more details, see [13].

The image, under a perspective projection, of a quadric L is a conic l. This relation is expressed by

$$\lambda l = P L P^T \ , \tag{9}$$

where P is the camera matrix and λ a scale factor. Introducing

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

and specializing (9) to an affine camera matrix gives

$$\lambda \begin{bmatrix} l_1 & l_2 & l_4 \\ l_2 & l_3 & l_5 \\ l_4 & l_5 & l_6 \end{bmatrix} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} L_1 & L_2 & L_4 & L_7 \\ L_2 & L_3 & L_5 & L_8 \\ L_4 & L_5 & L_6 & L_9 \\ L_7 & L_8 & L_9 & L_{10} \end{bmatrix} \begin{bmatrix} A^T & 0 \\ b^T & 1 \end{bmatrix} \ . \tag{11}$$

These equations can be divided into two sets of equations. The first set is

$$\lambda \begin{bmatrix} l_1 & l_2 \\ l_2 & l_3 \end{bmatrix} = A \begin{bmatrix} L_1 & L_2 & L_4 \\ L_2 & L_3 & L_5 \\ L_4 & L_5 & L_6 \end{bmatrix} A^T + A \begin{bmatrix} L_7 & L_8 & L_9 \end{bmatrix}^T b^T + b \begin{bmatrix} L_7 & L_8 & L_9 \end{bmatrix} A^T + b L_{10} b^T \ , \tag{12}$$

containing three nonlinear equations in A and b. The second set is

$$\lambda \begin{bmatrix} l_4 \\ l_5 \\ l_6 \end{bmatrix} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} L_7 \\ L_8 \\ L_9 \\ L_{10} \end{bmatrix} \ , \tag{13}$$

containing three linear equations in A and b.

Normalising l such that $l_6 = 1$ and L such that $L_{10} = 1$, the linear equations in (13) can be written

$$\begin{bmatrix} l_4 \\ l_5 \end{bmatrix} = A \begin{bmatrix} L_7 \\ L_8 \\ L_9 \end{bmatrix} + b \ . \tag{14}$$

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 nonlinear equations in (12) are omitted. The geometrical interpretation of (14) is that the center of the quadric projects onto the center of the conic in the image, since indeed $[\, l_4/l_6 \, l_5/l_6 \,]^T$ corresponds to the center of the conic.

3 Three views

In the projective case, the trifocal tensor plays a fundamental role in reconstruction from three views. In this section, we derive the analogous tensor for the affine camera. Furthermore, we show how this tensor can be linearly estimated from points, lines and conics.

3.1 The affine quasi-tensor

We start by looking at points. Assume that relative coordinates are used and denote the three camera matrices by A, B and C. A point $\mathbf{X}$ is projected onto the three views as

$$\mathbf{x} = A\mathbf{X}, \quad \mathbf{x}' = B\mathbf{X} \quad \text{and} \quad \mathbf{x}'' = C\mathbf{X} \ ,$$

or equivalently

$$M \begin{bmatrix} \mathbf{X} \\ -1 \end{bmatrix} = \begin{bmatrix} A & \mathbf{x} \\ B & \mathbf{x}' \\ C & \mathbf{x}'' \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ -1 \end{bmatrix} = 0 \ . \tag{15}$$

From the above equation, it follows that rank $M \leq 3$ since the nullspace of M is non-empty and, in turn, this implies that all 4×4 minors of M vanish. There are in total $\binom{6}{4} = 15$ such minors and they are linear expressions in the coordinates of $\mathbf{x}$, $\mathbf{x}'$ and $\mathbf{x}''$. By using Laplace expansions, see [3], on these minors it can be seen that they are built up by sums of terms that are products of an image coordinate and a 3×3 minor formed by three rows from the camera matrices A, B and C. Let

$$T = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \ . \tag{16}$$

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

Definition 1. *The minors built up by the rows i, j and k from T in (16) will be called the* **affine quasi-tensor** *and its $\binom{6}{3} = 20$ components will be denoted by t_{ijk}.* ■

Observe that t_{ijk} is not a tensor, since the components transform differently depending on which rows that are selected from T. The components arising from the selection of one row from each of A, B and C can be reordered to constitute a tri-valent tensor that is contravariant in all indices, whereas the components arising from two rows from A and one row from C can be reordered into a two-valent tensor that is covariant in one index and contravariant in the other.

Given the image coordinates in all three images the minors obtained from M in (15) give linear constraints on the 20 components of the affine quasi-tensor. As an example, the minor obtained by picking the first, second, third and fifth row from M can be written

$$t_{123}x'' - t_{125}x' + t_{135}y - t_{235}x = 0 \ .$$

There are in total 15 such linear constraints on the components of the affine quasi- tensor. These can be written

$$Rt = 0 \ , \tag{17}$$

where R is a 15×20 matrix containing relative image coordinates of the image point and t is a vector containing the 20 components of the affine quasi-tensor. From (17), it follows that the overall scale of the tensor components can not be determined. This means that if t_{ijk} constitute the components of an affine quasi-tensor, then λt_{ijk}, where $0 \neq \lambda \in \mathbb{R}$, also constitute components of an affine quasi-tensor 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. Observe that since relative coordinates are used, one point alone gives no constraints on the tensor, because its relative coordinates are all zero. The number of linearly independent constraints for different number of point correspondences are given in the following proposition.

Proposition 1. *Two corresponding points in 3 images give 10 linearly independent constraints on the components of the affine quasi-tensor. Three points give 16 constraints and four or more points give 19 constraints. Thus the components of the affine quasi-tensor can be linearly recovered from at least four point correspondences in 3 images.*

Proof. Use MAPLE or some other symbolic system to compute the rank of the matrices obtained by stacking a different number of R:s as in (17) above each other.

The next question is how to calculate the camera matrices A, B and C from the 20 components of the affine quasi-tensor. Observe first that the camera matrices can never be recovered uniquely from the quasi-tensor, since a multiplication

by an arbitrary non-singular 3×3 matrix to the right of T in (16) only changes the common scale of the tensor components. Thus, a natural representation for the three camera matrices is to set

$$A = \begin{bmatrix} 1 & 0 & 0 \\ a_1 & a_2 & a_3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 & 0 \\ b_1 & b_2 & b_3 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 0 & 0 & 1 \\ c_1 & c_2 & c_3 \end{bmatrix} . \tag{18}$$

Using this parametrisation of the three camera matrices, the unknown parameters a_i, b_j and c_k can easily be calculated according to the following proposition.

Proposition 2. *Given an affine quasi-tensor normalised such that* $t_{135} = 1$, *the camera matrices in the form of* (18) *can be calculated as*

$$A = \begin{bmatrix} 1 & 0 & 0 \\ t_{235} & t_{125} & -t_{123} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 & 0 \\ -t_{345} & t_{145} & t_{134} \end{bmatrix} \quad \textit{and} \quad C = \begin{bmatrix} 0 & 0 & 1 \\ t_{356} & -t_{156} & t_{136} \end{bmatrix} .$$

Notice that only 9 of the 20 components of the affine quasi-tensor have been used to calculate the camera matrices. This indicates that the 20 components (defined up to a common scale factor) obey 10 polynomial constraints in order to form the components of the affine quasi-tensor. In fact, we have the following theorem.

Theorem 1. *The* 20 *numbers* t_{ijk}, *normalised such that* $t_{135} = 1$ *constitute the components of an affine quasi-tensor if and only if*

$$\begin{aligned}
t_{146} &= t_{136}t_{145} + t_{134}t_{156}, \quad t_{236} = t_{136}t_{235} + t_{123}t_{356}, \quad t_{245} = t_{145}t_{235} + t_{345}t_{125}, \\
t_{124} &= t_{125}t_{134} + t_{123}t_{145}, \quad t_{126} = t_{125}t_{136} - t_{123}t_{156}, \quad t_{234} = t_{235}t_{134} - t_{345}t_{123}, \\
t_{346} &= t_{345}t_{136} + t_{356}t_{134}, \quad t_{256} = t_{235}t_{156} + t_{356}t_{125}, \quad t_{456} = t_{356}t_{145} - t_{345}t_{156}, \\
t_{246} &= t_{345}(t_{136}t_{125} - t_{123}t_{156}) + t_{134}(t_{235}t_{156} + t_{356}t_{125}) + \\
&\qquad t_{145}(t_{136}t_{235} + t_{123}t_{356}).
\end{aligned}$$

Proof. The theorem follows immediately by calculating suitable minors from (16), using the camera matrices in Proposition 2.

We now turn to the use of line correspondences to constrain the components of the affine quasi-tensor. Consider (6) for three different images of a line with direction $\mathbf{D}$ in 3D space, i.e.

$$\lambda \mathbf{d} = A\mathbf{D}, \quad \lambda' \mathbf{d}' = B\mathbf{D} \quad \text{and} \quad \lambda'' \mathbf{d}'' = C\mathbf{D} . \tag{19}$$

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

$$N \begin{bmatrix} \mathbf{D} \\ -\lambda \\ -\lambda' \\ -\lambda'' \end{bmatrix} = \begin{bmatrix} A & \mathbf{d} & 0 & 0 \\ B & 0 & \mathbf{d}' & 0 \\ C & 0 & 0 & \mathbf{d}'' \end{bmatrix} \begin{bmatrix} \mathbf{D} \\ -\lambda \\ -\lambda' \\ -\lambda'' \end{bmatrix} = 0 . \tag{20}$$

Thus the nullspace of N is non-empty, hence $\det N = 0$. Developing this determinant, we see that it is a trilinear expression in $\mathbf{d}$, $\mathbf{d}'$ and $\mathbf{d}''$ with coefficients that are suitable minors of T in (16). The coefficients appearing here is a subset of the components of the affine quasi-tensor. The subset consists of the minors formed by picking one row from each of A, B and C, i.e. t_{ijk}, where $i \in \{1,2\}$, $j \in \{3,4\}$ and $k \in \{5,6\}$. This subset constitutes a tensor and it is the one used in [12]. 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 components of the affine quasi-tensor.

It is evident from (13) that conics can be used in the same way as points, when the nonlinear equations for conics are omitted. In this way each conic correspondence acts as a point correspondence and gives the same number of constraints on the viewing geometry.

3.2 The reduced affine quasi-tensor

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, cf. [18]. However, this comparison is not fair, because our 20 number describes *all* trilinear functions between three affine views and should be compared to the $3 \times 16 = 48$ and $3 \times 27 = 81$ components of all trilinear tensors between three affine views and three projective views, respectively. However, it is possible to use a tensor with only 12 components to describe the viewing geometry in our case.

In order to obtain a smaller number of parameters, start again from (15) and rank $M \leq 3$. This time we will only consider the 4×4 minors of M that contain both rows one and two, one of rows three and four, and one of rows five and six. There are in total 4 such minors and they are linear in the coordinates of $\mathbf{x}$, $\mathbf{x}'$ and $\mathbf{x}''$. Again, these linear expressions have coefficients that are 3×3 minors of T in (16), but this time the only minors occurring are the ones containing both rows from A and one from B and one from C.

Definition 2. *The minors built up by the rows i, j and k, where either $i \in \{1,2\}$, $j \in \{3,4\}$, $k \in \{5,6\}$ or $i = 1$, $j = 2$, $k \in \{3,4,5,6\}$, from T in (16) will be called the* **reduced affine quasi-tensor** *and its 12 components will be denoted by t_{ijk}.* ■

Observe once more that t_{ijk} is not a real tensor since different components transform differently.

Given the image coordinates in all three images, the minors of the chosen rows obtained from M give linear constraints on the 12 components of the reduced affine quasi-tensor. There are in total 4 such linear constraints on the components of the affine quasi-tensor. These can be written

$$R^r t^r = 0 \ , \tag{21}$$

where R^r is a 4×12 matrix containing relative image coordinates of the image point and t^r is a vector containing the 12 components of the reduced affine quasi-tensor. Observe again that the overall scale of the tensor components can not be determined. In the same manner as in the previous section, we can prove the following.

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

Again the camera matrices can be calculated from the 12 components of the reduced affine quasi-tensor. The parametrisation in (18) will be used again.

Proposition 4. *Given a reduced affine quasi-tensor normalised such that* $t_{135} = 1$*, the camera matrices in the form of* (18) *can be calculated as*

$$A = \begin{bmatrix} 1 & 0 & 0 \\ t_{235} & t_{125} & -t_{123} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 & 0 \\ (t_{145}t_{235} - t_{245})/t_{125} & t_{145} & (t_{136} - t_{123}t_{145})/t_{125} \end{bmatrix}$$

$$\text{and} \quad C = \begin{bmatrix} 0 & 0 & 1 \\ (t_{236} - t_{136}t_{235})/t_{123} & (t_{126} - t_{125}t_{136})/t_{123} & t_{136} \end{bmatrix} .$$

Theorem 2. *The* 12 *numbers* t_{ijk}*, normalised such that* $t_{135} = 1$ *constitute the components of a reduced affine quasi-tensor if and only if*

$$(t_{146} - t_{136}t_{145})t_{123}t_{125} = (t_{124} - t_{123}t_{145})(t_{126} - t_{125}t_{136}),$$
$$t_{246}t_{125}t_{123} = (t_{123}t_{245}t_{126} + t_{125}t_{124}t_{236} - t_{235}t_{124}t_{126}) .$$

Proof. Follows immediately by calculating suitable minors from (16), using the camera matrices in Proposition 4.

Corresponding lines and conics can be used in the same way as before to constrain the components of the reduced quasi-tensor.

Using these 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 nonlinear properties, given in Theorem 1 and Theorem 2, of 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 Many views

In the landmark paper [17] a factorisation method was presented for the orthographic camera using point features. In this section we generalize this algorithm for the affine camera using not only points, but lines and conics as well. The algorithm takes any number of points, lines and conics in any number of images as input and the result is a reconstruction of the scene structure as well as the camera motion.

4.1 The factorisation algorithm

In Section 2 we derived how points, lines and conics project onto the image. Now consider m points or conics, and n lines in p images. From equations (4) and (6) we see that this can be written as one single matrix equation (with relative coordinates),

$$S = \begin{bmatrix} \mathbf{x}_{11} & \dots & \mathbf{x}_{1m} & \lambda_{11}\mathbf{d}_{11} & \dots & \lambda_{1n}\mathbf{d}_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{p1} & \dots & \mathbf{x}_{pm} & \lambda_{p1}\mathbf{d}_{p1} & \dots & \lambda_{pn}\mathbf{d}_{pn} \end{bmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ A_p \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \dots & \mathbf{X}_m & \mathbf{D}_1 & \dots & \mathbf{D}_n \end{bmatrix} \quad . \quad (22)$$

The right-hand side of (22) is the product of a $2p \times 3$ matrix and a $3 \times (m+n)$ matrix, which gives the following theorem.

Theorem 3. *The matrix S in (22) must obey*

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

Observe that the matrix S contains entries obtained from measurements in the images, as well as the unknown scalar factors λ_{ij}, which have to be estimated. Assuming that these are known, the camera matrices, the 3D points and the 3D directions can be obtained by factorising 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 Σ is a diagonal matrix containing the singular values, σ_i, of S. Let $\tilde{\Sigma} = \operatorname{diag}(\sigma_1, \sigma_2, \sigma_3, 0, \dots, 0)$ and let $\tilde{U}$ and $\tilde{V}$ denote the first three columns of U and V, respectively. Then

$$\begin{bmatrix} A_1 \\ \vdots \\ A_p \end{bmatrix} = \tilde{U}\sqrt{\tilde{\Sigma}} \quad \text{and} \quad \begin{bmatrix} X_1 & \dots & X_m & D_1 & \dots & D_n \end{bmatrix} = \sqrt{\tilde{\Sigma}}\tilde{V}^T \quad (23)$$

fulfil (22). 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.

4.2 The initialisation

We now turn to the initialisation of the scalar factors, where the previously defined tensors will be useful. Assume that the (reduced) affine quasi-tensors have been calculated. Then the camera matrices can be calculated from Proposition 2 or Proposition 4. It follows from (20) that once the camera matrices for three images are known, the scalar factors for each direction can be calculated up to an unknown scalar factor. It remains to estimate the scalar 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 (20) as

$$M \begin{bmatrix} \mathbf{D} \\ -1 \end{bmatrix} = \begin{bmatrix} A_1 & \lambda_1\mathbf{d}_1 \\ A_2 & \lambda_2\mathbf{d}_2 \\ A_3 & \lambda_3\mathbf{d}_3 \end{bmatrix} \begin{bmatrix} \mathbf{D} \\ -1 \end{bmatrix} = 0 \ , \quad (24)$$

shows that M in (24) has rank less than 4 which implies that all 4×4 minors are equal to zero. These minors give linear constraints on the scale factors. However, only 3 of them are independent. So we get a system with the following appearance,

$$\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = 0 \ , \tag{25}$$

where $*$ indicates a matrix entry that can be calculated from A_i and $\mathbf{d}_i$. It is evident from (25) that the scalar factors λ_i only can be calculated up to an unknown common scale factor. By considering another triplet, with two images in common with the first triple, say the last two, we can obtain consistent scale factors for both triplets by solving a system with the following appearance,

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

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.

4.3 Summary

In summary, the following algorithm is proposed.

1. Calculate the scalar factors λ_{ij} using an affine quasi-tensor.
2. Calculate S in (22) from λ_{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 (23).
5. Reconstruct 3D lines and 3D quadrics.

The last step needs a further comment. From the factorisation the 3D directions of the lines and the centers of the quadrics are obtained. The remaining unknowns can be recovered linearly from (5) and (12).

If further accuracy is required, the reconstruction can be refined with bundle adjustment techniques.

5 Experiments

In this section we present some experimental results on the developed theory. The experiments have been performed on both synthetic and real data.

5.1 Simulated Data

The synthetic data was produced in the following. First, points and line segments were randomly distributed within a sphere. Then, views corresponding to camera positions on the sphere were randomly chosen and the features were projected to these views. In each view, the coordinate system was chosen so that the data lies within the range $[-1, 1]$ to improve numerical conditioning. 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 segment were perturbed with independent Gaussian noise in the normal direction of the line. Then, the line parameters were estimated with least-squares. The residual error for points was chosen as the distance between the true point position and the reprojected reconstructed 3D point. For lines, the residual errors were chosen as the smallest distances between the endpoints of the true line segment and the reprojected 3D line. These settings are close to real life situations (up to scale). A noise level of 0.005 corresponds approximately to a perturbation of 1 pixel for a 512×512 image.

STD of noise	0	0.005	0.01	0.02
Red. quasi-tensor				
RMS of points	0.0	0.015	0.030	0.068
RMS of lines	0.0	0.012	0.026	0.057
Quasi-tensor				
RMS of points	0.0	0.0045	0.0086	0.017
RMS of lines	0.0	0.0031	0.0060	0.013
Factorisation				
RMS of points	0.0	0.0043	0.0088	0.017
RMS of lines	0.0	0.0029	0.0060	0.012

Table 1. Result of simulations of 10 points and 10 lines in 3 images for different levels of noise using the affine quasi-tensor, the reduced affine quasi-tensor and the factorisation approach. The root mean square (RMS) errors are shown for the different approaches.

In Table 1 it can be seen that the 20-parameter formulation of the three views is consistently superior to the 12-parameter formulation. For three views, nothing is gained by applying the factorisation method. All three methods handle moderate noise perturbations well. In Table 2 the number of points and lines are varied. The more points and lines used the better results as expected. Finally, in Table 3 the number of views is varied. In spite of the rather high noise level, the factorisation method manages to keep the residuals low.

5.2 Real Data

The presented methods have been tested on real data as well. In this section an experiment is presented that was performed on an outdoor statue, which is

#points, #lines	3,3	10,10	20,20	30,30
Red. quasi-tensor				
RMS of points	0.0094	0.030	0.024	0.021
RMS of lines	0.024	0.026	0.018	0.015
Quasi-tensor				
RMS of points	0.0094	0.0086	0.0082	0.0080
RMS of lines	0.021	0.0060	0.0055	0.0053
Factorisation				
RMS of points	0.0091	0.0088	0.0081	0.0079
RMS of lines	0.028	0.0060	0.0054	0.0054

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 0.05. The table shows the resulting error (RMS) after using the differnt approaches.

#views	3	5	10	20
Factorisation				
RMS of points	0.017	0.019	0.019	0.026
RMS of lines	0.013	0.015	0.016	0.022

Table 3. Table showing simulated results for 10 points and 10 lines in a different number of views, with an added error of standard deviation 0.02 .

in fact built up by conics and lines. 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(a).

In the experiment, 5 different images (768×575) of the statue were taken. In these images, the two ellipses, 17 lines and 17 points were picked out by hand and for the ellipses and lines, least-squares were used to compute the appropriate representations. The residual errors are shown in Table 4 in pixels with the same definitions of residual errors (between measured and reprojected quantities) as in the previous experiments. The results are clearly plausible, see Figure 1(b).

Factorisation	Points	Lines	Conics(center)
RMS pixels	4.3	0.56	6.6

Table 4. Table showing the result of statue experiment with five real images.

6 Conclusions

In this paper a novel scheme that can handle any number of corresponding points, lines and conics in any number of images, taken by affine cameras has been presented. Two novel concepts have been introduced; the affine quasi-tensor

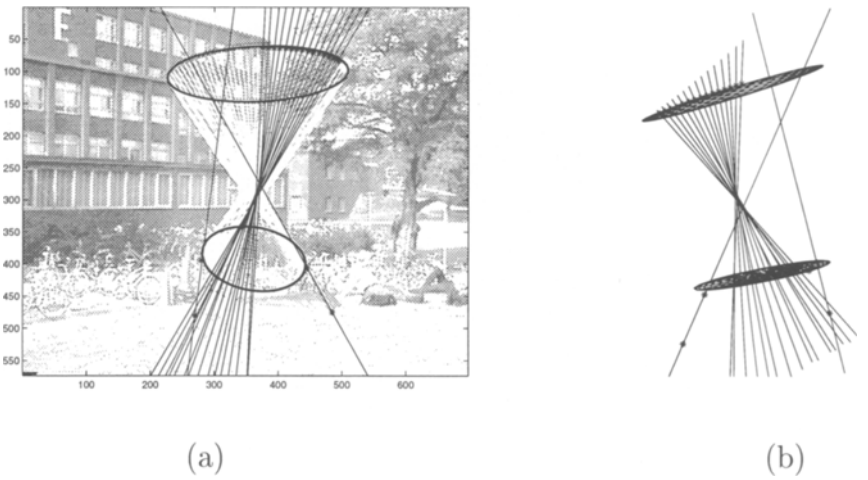

Fig. 1. (a) Image of a statue. (b) Reconstructed 3D model

and the reduced affine quasi-tensor. Using these concepts, the camera matrices in a triplet of affine views can be estimated linearly from corresponding points, lines and conics. First, the tensor components (20 or 12) are estimated from image data and then the camera matrices are obtained from the tensor components. A slight generalization of this procedure gives also the scale factors for all line directions, needed to initialise the factorisation method. Furthermore, it has been shown that this approach works well on both simulated and real data.

References

1. R. Berthilsson and K. Åström. Reconstruction of 3d-curves from 2d-images using affine shape methods for curves. In *Proc. Conf. Computer Vision and Pattern Recognition*, 1997.
2. O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, MA*, pages 951–956, 1995.
3. B. Gelbaum. *Linear Algebra : Basic, Practice and Theory.* North-Holand, 1989.
4. R. I. Hartley. A linear method for reconstruction from lines and points. In *Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, MA*, pages 882–887, 1995.
5. R. I. Hartley. Lines and points in three views and the trifocal tensor. *Int. Journal of Computer Vision*, 22(2):125–140, 1997.
6. A. Heyden. Projective structure and motion from image sequences using subspace methods. In *Proc. 10th Scandinavian Conf. on Image Analysis, Lappeenranta, Finland*, pages 963–968, 1997.
7. F. Kahl and K. Åström. Motion estimation in image sequences using the deformation of apparent contours. In *Proc. 6th Int. Conf. on Computer Vision, Mumbai, India*, 1998.

8. F. Kahl and A. Heyden. Using conic correspondences in two images to estimate the epipolar geometry. In *Proc. 6th Int. Conf. on Computer Vision, Mumbai, India*, 1998.
9. J. L. Mundy and A. Zisserman, editors. *Geometric invariance in Computer Vision.* MIT Press, Cambridge Ma, USA, 1992.
10. T. Papadopoulo and O. Faugeras. Computing structure and motion of general 3d curves from monocular sequences of perspective images. In B Buxton and R. Cipolla, editors, *Proc. 4th European Conf. on Computer Vision, Cambridge, UK*, pages 696–708. Springer-Verlag, 1996.
11. L. Quan. Conic reconstruction and correspondence from two views. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(2):151–160, February 1996.
12. L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(8), August 1997.
13. J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry.* Clarendon Press, Oxford, 1952.
14. L. S. Shapiro. *Affine Analysis of Image Sequences.* Cambridge University Press, 1995.
15. G. Sparr. Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In *Proc. Int. Conf. on Pattern Recognition, Vienna, Austria*, 1996.
16. P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proc. 4th European Conf. on Computer Vision, Cambridge, UK*, pages 709–720, 1996.
17. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. Journal of Computer Vision*, 9(2):137–154, 1992.
18. P. Torr. *Motion Segmentation and Outlier Detection.* PhD thesis, Department of Engineering Science, University of Oxford, 1995.
19. B. Triggs. Matching constraints and the joint image. In *Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, MA*, pages 338–343, 1995.
20. J. Weng, T.S. Huang, and N. Ahuja. Motion and structure from line correspondances: Closed-form solution, uniqueness, and optimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(3), 1992.