

Structure and Motion From Lines Under Affine Projections*

Kalle Åström, Anders Heyden, Fredrik Kahl, Magnus Oskarsson,
Centre for Mathematical Sciences, Lund University
Box 118, S-221 00 Lund, Sweden
{kalle,heyden,fredrik,magnuso}@maths.lth.se

Abstract

In this paper we investigate the geometry and algebra of multiple projections of lines with affine cameras. Previously, the case of seven lines in three images has been studied. It was thought that this was the minimal data necessary for recovering affine structure and motion and that there are in general two solutions. It was also thought that these two solutions persist with more than seven lines. In this paper it is shown that the minimal cases are six lines in three images and five lines in four images. These cases are solved and it is shown that there are in general four solutions in both problems. Two almost minimal cases (seven lines in three images and six lines in four images) are solved using linear methods. Furthermore, it is shown that the solution is in general unique in these almost minimal cases. Finally, experiments are conducted on both simulated and real data in order to show the applicability of the theory.

1. Introduction

One key problem in computer vision is the ability to simultaneously calculate the three-dimensional scene structure and the position and orientation of the camera using only image features, see [2]. This, so called structure and motion problem, divides into several different subproblem depending on the chosen camera model (perspective, affine, orthographic, etc.) and type of image features (points, lines, conics, general curves, etc.). In this paper we study the structure and motion problem for lines under the so called affine camera. The affine camera model, first introduced in [13], is a generalization of the orthographic, the weak perspective and the para-perspective camera models. Several algorithms exist for recovering structure and motion, mostly for points [17, 9], but also for other image features [10, 11, 15].

In [14], it was shown that the structure and motion problem using line features in affine cameras can be reduced to the structure and motion problem for points in one dimension less, i.e. for one-dimensional cameras. This simplification was obtained by considering the direction of the lines and neglecting the actual position and was used to solve the problem of three views of seven lines. It was claimed that this is the minimal data necessary and that two solutions are obtained. In this paper we show that although two solutions can be obtained for the reduced problem (2D points projects to 1D retina using only directions), only one of the solutions can be used for the original problem, i.e. when also the position of the lines are taken into account.

Furthermore, it is shown how to solve the case of 6 lines in 4 images using the same idea of first studying the line directions as 2D points projected onto a 1D retina. New results on a dual hexalinear tensor (for 1D retina) makes it possible to estimate this almost minimal case linearly. These results can be used in other contexts as well. Also in this case there are two solutions to the reduced problem but only one for the original problem.

In contrast to the results in [14] we show that seven points in three views is not a minimal case. The minimal cases for structure and motion for lines are 5 points in 4 images and 6 points in 3 images. In order to solve these cases, we derive the necessary and sufficient conditions on the trifocal and quadrifocal tensors under the assumption of affine cameras. It is shown that each line triplet gives two constraints on the trifocal tensor. This makes it possible to solve the case of 6 lines in 3 images and it turns out that there are in general 4 solutions. Moreover, each line quadruple gives 9 constraint on the quadrifocal tensor. This makes it possible to solve the case of 5 lines in 4 images. There are in general 4 solutions in this case also.

The results presented here are of both practical and theoretical interest. The solution to the minimal cases can be used as initialization for bundle adjustment routines that calculates maximum likelihood estimates of structure and motion. Minimal cases are also useful in robust estimation algorithms like RANSAC or LMS [18, 20] for finding cor-

*This work has been done within then ESPRIT Reactive LTR project 21914, CUMULI and the Swedish Research Council for Engineering Sciences (TFR), project 95-64-222

respondences. The linear algorithms can be used when redundant lines are available to get unique solutions and computationally cheap estimates.

2. Enumerative geometry

In this paper we will use the affine camera model, i.e. points with coordinates (X, Y, Z) are projected to image points with coordinates (x, y) according to

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & p_{34} \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} .$$

The camera matrix \mathbf{P} represents the position and orientation of the camera. It is only defined up to scale and it is convenient to assume that the scale of \mathbf{P} is chosen so that $p_{34} = 1$.

Lines can be represented in many ways. One way is to specify one point on the line and the direction of the line. For 3D lines we get

$$\mathbf{L} = \left\{ \mathbf{U} + s \begin{bmatrix} \mathbf{D} \\ 0 \end{bmatrix} \mid s \in \mathbb{R} \right\} ,$$

where \mathbf{U} denotes a point and \mathbf{D} a direction, and for image lines:

$$\mathbf{l} = \left\{ \mathbf{u} + s \begin{bmatrix} \mathbf{d} \\ 0 \end{bmatrix} \mid s \in \mathbb{R} \right\} ,$$

where \mathbf{u} and \mathbf{d} denotes a point on the line and the direction respectively. Inserting these line representations in the affine camera model gives

$$\mathbf{P} \left(\mathbf{U} + s \begin{bmatrix} \mathbf{D} \\ 0 \end{bmatrix} \right) = \mathbf{P}\mathbf{U} + s \begin{bmatrix} \mathbf{P}_{2 \times 3} \mathbf{D} \\ 0 \end{bmatrix} = \mathbf{u} + s \begin{bmatrix} \mathbf{d} \\ 0 \end{bmatrix} , \quad (1)$$

where $\mathbf{P}_{2 \times 3}$ is the upper left 2×3 sub-matrix of \mathbf{P} . Notice that the projection of **line directions** is

$$\mu \mathbf{d} = \mathbf{P}_{2 \times 3} \mathbf{D} . \quad (2)$$

This can be seen as a projection of a 2D point to a 1D retina using a general projective camera. This was used in [14] for the solution of 7 lines in 3 images. Some information is, however, lost in this simplification. The condition that the directions fulfils (2) is only a *necessary* condition for the affine reconstruction problem using lines, not a sufficient condition, since the position of the lines are not taken into account.

Another representation for image lines is the so called dual coordinates. In this case a line is represented by three homogeneous coordinates $\mathbf{l} = [a \ b \ c]^T$ and the line is given by $\mathbf{l}^T \mathbf{u} = 0$, where $\mathbf{u} = [x \ y \ 1]^T$ and (x, y) denote points on the line. This representation is useful in many situations.

| m | n | | | | | |
|---|----|----|----|-----|-----|-----|
| | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | -4 | -6 | -8 | -10 | -12 | -14 |
| 2 | -4 | -4 | -4 | -4 | -4 | -4 |
| 3 | -4 | -2 | 0 | 2 | 4 | 6 |
| 4 | -4 | 0 | 4 | 8 | 12 | 16 |
| 5 | -4 | 2 | 8 | 14 | 20 | 26 |

Table 1. The number of excess constraints $2mn - (4n + 8m - 12)$ for the structure and motion problem with m images of n points.

For example the problem of determining the 3D line from its image lines \mathbf{l}^i , where i denotes image number, with known camera matrices \mathbf{P}_i (so called **intersection**) can be solved by intersecting the planes $\mathbf{P}_i^T \mathbf{l}^i$. A necessary condition for these planes to meet in a common line is

$$\text{rank} [\mathbf{P}_1^T \mathbf{l}^1 \ \dots \ \mathbf{P}_m^T \mathbf{l}^m] = 2 ,$$

i.e. each 3×3 minor of the $4 \times m$ matrix above vanish. These constraints can be formulated in terms of elements of the so called trifocal and quadrifocal tensors (see Sections 3 and 5).

In order to understand how much information is needed in order to solve the structure and motion problem, it is useful to calculate the number of degrees of freedom of the unknown parameters and the number of constraints given by the projection equation. Each line has four degrees of freedom and each camera has eight. The solution is only defined up to an affine transformation, which has dimension 12. Using n lines and m cameras we thus have $4n + 8m - 12$ degrees of freedom in the parameter space. Each measured line gives two constraints on the estimated parameters. Assuming that each line is visible in every image we get $2mn$ constraints. The number of excess constraints $2mn - (4n + 8m - 12)$ is shown in Table 1. Note that there are two interesting minimal cases.

1. Three images of six lines ($m = 3, n = 6$).
2. Four images of five lines ($m = 4, n = 5$).

In Sections 3 and 5, these cases are analyzed in detail.

By considering line directions only, the problem is transformed to projections of 2D points to a 1D retina, cf. (2). Using n points and m cameras we thus have $2n + 5m - 8$ degrees of freedom in the parameters. Each measured direction gives one constraint on the estimated parameters. Assuming that each point is visible in every camera we get mn constraints. The number of excess constraints $mn - (2n + 5m - 8)$ is shown in Table 2. Note that here the minimal cases are

1. Three images of seven lines ($m = 3, n = 7$).

| m | n | | | | | |
|---|----|----|----|----|----|----|
| | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | -1 | -2 | -3 | -4 | -5 | -6 |
| 2 | -2 | -2 | -2 | -2 | -2 | -2 |
| 3 | -3 | -2 | -1 | 0 | 1 | 2 |
| 4 | -4 | -2 | 0 | 2 | 4 | 6 |
| 5 | -5 | -2 | 1 | 4 | 7 | 10 |

Table 2. The number of excess constraints $mn - (2n + 5m - 8)$ for the structure and motion problem with m images of n points.

2. Four images of six lines ($m = 4, n = 6$).

We will show how both these cases can be solved with linear algorithms, see Sections 4 and 6.

3. 6 lines in 3 images

Consider three affine cameras ($\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$) and three corresponding lines, ($\mathbf{l}^1, \mathbf{l}^2, \mathbf{l}^3$). The condition that these three lines are images of a common 3D line is

$$\text{rank} [\mathbf{P}_1^T \mathbf{l}^1 \quad \mathbf{P}_2^T \mathbf{l}^2 \quad \mathbf{P}_3^T \mathbf{l}^3] = 2 .$$

Expanding the 3×3 minors, the constraints can be written (using Einsteins' summation convention)

$$\mathbf{l}_i^1 \sim T_i^{jk} \mathbf{l}_j^2 \mathbf{l}_k^3 , \quad (3)$$

where \sim denotes equality up to scale and T_i^{jk} denotes the elements of the trifocal tensor, which is defined as

$$T_i^{jk}(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3) = \varepsilon_{i'j'k'} \det \begin{bmatrix} \mathbf{P}_1^{i'} \\ \mathbf{P}_1^{j'} \\ \mathbf{P}_2^j \\ \mathbf{P}_3^k \end{bmatrix} ,$$

where ε denotes the permutation tensor. The necessary and sufficient conditions for the trifocal tensor for projective cameras are given in [4, 3, 8]. For affine cameras these conditions can be simplified a little.

Theorem 3.1. *Necessary and sufficient conditions for an affine trifocal tensor are*

$$\begin{aligned} T_1^{13} = 0, T_1^{23} = 0, T_1^{33} = 0, T_1^{32} = 0, \\ T_1^{31} = 0, T_2^{13} = 0, T_2^{23} = 0, T_2^{33} = 0, \\ T_2^{32} = 0, T_2^{31} = 0, T_3^{33} = 0, \end{aligned}$$

$$\begin{aligned} T_3^{11} T_3^{23} T_3^{32} - T_3^{21} T_3^{13} T_3^{32} - T_3^{31} T_3^{12} T_3^{23} + T_3^{31} T_3^{13} T_3^{22} = 0, \\ T_2^{11} T_3^{23} T_3^{32} - T_2^{21} T_3^{13} T_3^{32} - T_3^{31} T_2^{12} T_3^{23} + T_3^{31} T_3^{13} T_2^{22} = 0, \\ T_1^{11} T_3^{23} T_3^{32} - T_1^{21} T_3^{13} T_3^{32} - T_3^{31} T_1^{12} T_3^{23} + T_3^{31} T_3^{13} T_1^{22} = 0 . \end{aligned} \quad (4)$$

Every line correspondence in three images gives two linear constraints on the trifocal tensor from (3). With 6 lines we thus get 12 linear constraints from the image data and 11 additional linear constraints from (4). This makes it possible to parameterize the tensor components using only four ($27 - 12 - 11 = 4$) parameters defined up to scale. Inserting this parameterization into the three third degree polynomial constraints in (4) gives up to 27 solutions counted with multiplicity. However, after a more careful examination, only a few of these solutions remain.

Theorem 3.2. *There are in general 4 algebraic solutions to the problem of recovering structure and motion for 6 lines affinely projected into 3 images. These may include complex and not physically feasible solutions.*

Proof. Without loss of generality, one can choose coordinate systems in the images such that the first two lines in each image have coordinates $(1, 0, 0)$ and $(0, 1, 0)$ respectively, which implies that (3) gives for the first line

$$T_2^{11} = T_3^{11} = 0$$

and for the second line

$$T_1^{22} = T_3^{22} = 0 .$$

The components $T_1^{11}, T_2^{22}, T_1^{12}, T_2^{12}, T_1^{21}, T_2^{21}, T_3^{12}$ and T_3^{21} can be eliminated linearly using the remaining four lines. This leaves us with the unknowns $T_3^{31}, T_3^{32}, T_3^{13}$ and T_3^{23} and the three non-linear constraints in (4). Using resultants between every pair of equations (with a computer algebra system like Maple), T_3^{31} and T_3^{32} can be eliminated. After removing spurious solutions (arising because of our particular parameterization of the problem), we end up with a homogeneous polynomial equation of degree four in T_3^{13} and T_3^{23} . The roots of this polynomial equation correspond to true solutions. ■

Algorithm 3.1. (Six lines in three affine views).

1. Calculate the trifocal tensor up to a four-parameter family of solutions, using the special choice of image coordinates as above.
2. Eliminate three of the parameters using the necessary conditions in Theorem 3.1.
3. Solve the fourth degree polynomial equation and substitute this solution back to get the trifocal tensor.
4. Calculate the camera matrices from the trifocal tensor.

4. 7 lines in 3 images

The structure and motion problem for 7 lines in 3 images, was solved in [14] using the projection of the line directions.

For completeness an outline of this method is presented here as well. However, it was claimed that there are two different solutions, which is true when only the directions of the lines are considered. We will show that when also the positions of the lines are taken into account, only one solutions will, in general, remain.

Definition 4.1. The group \mathcal{S} of planar projective transformations is defined as the set of equivalence classes of non-singular 3×3 matrices, where two matrices are considered equal if one is a scalar multiple of the other.

Definition 4.2. Let the manifold of **relative orientation** of three 1D cameras be defined as the set of equivalence classes of three ordered 1D camera matrices

$$\mathcal{P} = \left\{ (\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3) \mid \mathbf{P}_i = \begin{bmatrix} a_i & b_i & c_i \\ d_i & e_i & f_i \end{bmatrix} \right\} / \simeq \quad (5)$$

where the equivalence \simeq is defined as $(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3) \simeq (\tilde{\mathbf{P}}_1, \tilde{\mathbf{P}}_2, \tilde{\mathbf{P}}_3)$, if and only if

$$\exists \mathbf{s} \in \mathcal{S}, \quad \tilde{\mathbf{P}}_i \sim \mathbf{P}_i \mathbf{s}, i = 1, 2, 3 .$$

Definition 4.3. The manifold of two-dimensional trifocal tensors \mathcal{T} is defined as the equivalence class of $2 \times 2 \times 2$ tensors, where two tensors are considered equal if one is a multiple of the other.

Theorem 4.1. Let \mathbf{d}_1 , \mathbf{d}_2 and \mathbf{d}_3 denote three corresponding image line directions. Then the trilinear constraint

$$\sum_{i,j,k} T_{ijk} \mathbf{d}_1^i \mathbf{d}_2^j \mathbf{d}_3^k = 0 , \quad (6)$$

is fulfilled for the $2 \times 2 \times 2$ tensor T , defined by

$$T_{ijk} = \wedge_{i'i'} \wedge_{j'j'} \wedge_{k'k'} \det \begin{bmatrix} \mathbf{P}_1^{i'} \\ \mathbf{P}_2^{j'} \\ \mathbf{P}_3^{k'} \end{bmatrix} , \quad (7)$$

where $\wedge_{i'i'}$ denote the wedge operator

$$\wedge = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} .$$

Proof. By lining up the equations for projecting a line direction \mathbf{D}_j (1)

$$\underbrace{\begin{bmatrix} \mathbf{P}_1 & \mathbf{d}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{P}_2 & \mathbf{0} & \mathbf{d}_2 & \mathbf{0} \\ \mathbf{P}_3 & \mathbf{0} & \mathbf{0} & \mathbf{d}_3 \end{bmatrix}}_M \begin{bmatrix} \mathbf{D}_j \\ -\lambda_1 \\ -\lambda_2 \\ -\lambda_3 \end{bmatrix} = \mathbf{0} \quad (8)$$

we see that the 6×6 matrix M has a non-trivial nullspace and therefore its determinant is zero. Since the determinant is linear in each column it follows that it can be written

$$\det M = \sum_{i,j,k} T_{ijk} \mathbf{d}_1^i \mathbf{d}_2^j \mathbf{d}_3^k = 0 . \quad (9)$$

■

It is straightforward to calculate the tensor components from the camera matrices. However, there is a two-fold ambiguity when going in the other direction.

Theorem 4.2. For each tensor $T \in \mathcal{T}$ there are two possible motions in \mathcal{P} .

These theorems can be used to solve the 7 lines in 3 images problem as was done in [14]. However, when going from line directions to the original problem only one of the two possible solutions survive.

Another way to solve the problem is to use the affine trifocal tensor from the previous section. Each line triplet gives two constraints (3) on the 16 non-zero tensor components. Thus with 7 lines it is possible to solve for the tensor components up to a projective parameter:

$$T(x, y) = T_x x + T_y y, \quad (x, y) \in \mathbb{P}^1 .$$

The parameter $(x, y) \in \mathbb{P}^1$ can be obtained using the three non-linear constraints (4). It is then straightforward to compute the three camera matrices uniquely from T and then to compute the structure of the lines using resection. Since there are three non-linear constraints on the two homogeneous parameters, there is in general only one solution (assuming consistent data, i.e. that at least one solution exists). This is in contrast with the results presented in [14] where it is claimed that there are two solutions to the problem.

Algorithm 4.1. (\geq seven lines in three affine views).

1. Use the directions of the lines to calculate the two-dimensional trifocal tensor.
2. Calculate the relative orientation of the three 1D cameras up to a two-fold ambiguity.
3. Resolve the ambiguity by considering also the positions of the lines.

5. 5 lines in 4 images

Consider four affine cameras, $(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4)$ and the ordinary quadrifocal tensor

$$Q^{ijkl}(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4) = \det \begin{bmatrix} \mathbf{P}_1^i \\ \mathbf{P}_2^j \\ \mathbf{P}_3^k \\ \mathbf{P}_4^l \end{bmatrix} ,$$

where \mathbf{P}_1^i denotes the i th row of \mathbf{P}_1 etc., see [8, 19]. Necessary conditions for the 81 components of the quadrifocal tensor for projective cameras are given in [8].

Theorem 5.1. *The following conditions are necessary for the quadrifocal tensor:*

$$\det Q^{IJ \times \times} = \det \begin{bmatrix} Q^{IJ11} & Q^{IJ12} & Q^{IJ13} \\ Q^{IJ21} & Q^{IJ22} & Q^{IJ23} \\ Q^{IJ31} & Q^{IJ32} & Q^{IJ33} \end{bmatrix} = 0, \quad (10)$$

where $I, J = 1, 2, 3$, including all permutations of the indices.

For affine cameras it is easy to see that

$$Q^{ijkl} = 0 \quad \text{when at least two of } i, j, k, l \text{ are } 3.$$

Thus we 'only' have to deal with 48 non-zero components.

For each line correspondence, $\mathbf{l}^1, \mathbf{l}^2, \mathbf{l}^3, \mathbf{l}^4$, we get the following constraints on Q

$$\begin{aligned} Q^{ijkl} \mathbf{l}_i^1 \mathbf{l}_j^2 \mathbf{l}_k^3 &= 0, & Q^{ijkl} \mathbf{l}_i^1 \mathbf{l}_j^2 \mathbf{l}_l^4 &= 0, \\ Q^{ijkl} \mathbf{l}_i^1 \mathbf{l}_k^3 \mathbf{l}_l^4 &= 0, & Q^{ijkl} \mathbf{l}_j^2 \mathbf{l}_k^3 \mathbf{l}_l^4 &= 0, \end{aligned} \quad (11)$$

where \mathbf{l}_i^1 , $i = 1, 2, 3$ denote the three (dual) coordinates of the line \mathbf{l}^1 etc. Although (11) contains 12 equations it turns out that only 9 are linearly independent in the tensor components. To sum up we have $5 \times 9 = 45$ linear constraints from 5 line correspondences in the 48 tensor components. This means that we need two further constraints to calculate the quadrifocal tensor, which means that we have to use the non-linear necessary conditions in (10).

To simplify the calculations, we may choose new affine coordinate systems in the images such that the three first lines in each image have the coordinates $(1, 0, 0)$, $(0, 1, 0)$ and $(1, 1, 1)$ and further normalize the dual line coordinates such that the fourth and fifth line have its third coordinate equal to 1. Using these coordinates in the constraints in (11) it turns out that

$$Q^{ijkl} = 0 \quad \text{when at least three of } i, j, k, l \text{ are } 1 \text{ or } 2,$$

which implies that we now 'only' have 30 non-zero components of the quadrifocal tensor. This simplification is not necessary to prove the assertion on the number of solutions neither to calculate the different solutions numerically, but it decreases the complexity of the coefficients in the subsequent polynomial equations considerably.

The constraints in (11) also gives 27 linearly independent constraints on the tensor components. Using these constraints it is possible to constrain the non-zero tensor components to a three-dimensional linear subspace. This can explicitly be done by pre-selecting three components, e.g. Q^{3211} , Q^{3212} and Q^{3221} , and expressing the remaining tensor components as linear combinations of these. Considering the necessary condition $\det Q^{\times 2 \times 1} = 0$ gives a third order homogeneous polynomial constraint in these three

components, where the coefficients are polynomial expressions in the coordinates of the fourth and fifth line. Similarly considering the necessary conditions $\det Q^{\times 21 \times} = 0$, $\det Q^{\times \times 12} = 0$ and $\det Q^{\times \times 21} = 0$ gives three further third order homogeneous polynomial constraint. Taking resultants between these third order polynomials, finding the least common multiple and eliminating spurious solutions, gives finally a fourth degree polynomial constraint in one parameter.

Algorithm 5.1. (Five lines in four affine views).

1. Calculate the quadrifocal tensor up to a three-parameter family of solutions.
2. Eliminate two out of the three parameters using the necessary conditions in Theorem 5.1.
3. Solve the fourth degree polynomial equation and substitute this solution back to get the quadrifocal tensor.
4. Calculate the camera matrices from the quadrifocal tensor, see [8, 5].

6. 6 lines in 4 images

Below follows a method for reconstructing 6 lines from 4 images. The method is dual (in the sense of [1]) to the one of solving for 7 lines in 3 images as described in Section 4.

The projection equations for line directions can be written as

$$\lambda_{ij} \mathbf{d}_{ij} = \mathbf{P}_i \mathbf{D}_j = B(\mathbf{D}_j) \mathbf{p}_i$$

where

$$B \left(\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \right) = \begin{bmatrix} X & Y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & X & Y & 1 \end{bmatrix}$$

and

$$\mathbf{p}_i = [a_i \quad b_i \quad c_i \quad d_i \quad e_i \quad f_i]^T,$$

cf. (5). For a specific image i the projection equations for all six line directions in the image can be rewritten as (with image number omitted)

$$\underbrace{\begin{bmatrix} B(\mathbf{D}_1) & \mathbf{d}_1 & 0 & 0 & 0 & 0 \\ B(\mathbf{D}_2) & 0 & \mathbf{d}_2 & 0 & 0 & 0 \\ B(\mathbf{D}_3) & 0 & 0 & \mathbf{d}_3 & 0 & 0 \\ B(\mathbf{D}_4) & 0 & 0 & 0 & \mathbf{d}_4 & 0 \\ B(\mathbf{D}_5) & 0 & 0 & 0 & 0 & \mathbf{d}_5 \\ B(\mathbf{D}_6) & 0 & 0 & 0 & 0 & \mathbf{d}_6 \end{bmatrix}}_M \begin{bmatrix} \mathbf{p}_i \\ -\lambda_1 \\ -\lambda_2 \\ -\lambda_3 \\ -\lambda_4 \\ -\lambda_5 \\ -\lambda_6 \end{bmatrix} = \mathbf{0} \quad (12)$$

Observe that the 12×12 matrix M has a non-trivial nullspace, implying that its determinant is zero. Since the determinant is linear in each column it follows that it can be written as

$$\det M = \sum_{i,j,k,l,m,n} H_{ijklmn} \mathbf{d}_1^i \mathbf{d}_2^j \mathbf{d}_3^k \mathbf{d}_4^l \mathbf{d}_5^m \mathbf{d}_6^n = 0, \quad (13)$$

for a $2 \times 2 \times 2 \times 2 \times 2 \times 2$ tensor H . It is straightforward to see that the tensor components can be obtained as determinants of the first four rows of M . The notation $B(\mathbf{D}_j)^1$ and $B(\mathbf{D}_j)^2$ will be used for the two rows of the matrix $B(\mathbf{D}_j)$. The tensor components can be computed from the object points $(\mathbf{D}_1, \dots, \mathbf{D}_6)$ as

$$H((\mathbf{D}_1, \dots, \mathbf{D}_6))_{ijklmn} = \wedge_{i'l} \cdots \wedge_{m'n'} \det \begin{bmatrix} B(\mathbf{D}_1)^{i'} \\ B(\mathbf{D}_2)^{j'} \\ B(\mathbf{D}_3)^{k'} \\ B(\mathbf{D}_4)^{l'} \\ B(\mathbf{D}_5)^{m'} \\ B(\mathbf{D}_6)^{n'} \end{bmatrix} . \quad (14)$$

Theorem 6.1. Let $\mathbf{d}_{i,j}$, $j = 1, \dots, 6$ denote the line directions for six different lines in the same image. Then the dual hexalinear constraint (13) is fulfilled for a $2 \times 2 \times 2 \times 2 \times 2 \times 2$ tensor H , defined by (14).

Definition 6.1. Let the manifold \mathcal{U} of projective shape of six line directions be defined as the set of equivalence classes of six ordered line directions

$$\mathcal{U} = \left\{ (\mathbf{D}_1, \dots, \mathbf{D}_6) \mid \mathbf{D}_j \sim \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} \right\} / \simeq$$

where the equivalence \simeq is defined as $(\mathbf{D}_1, \dots, \mathbf{D}_6) \simeq (\tilde{\mathbf{D}}_1, \dots, \tilde{\mathbf{D}}_6)$ if and only if

$$\exists s \in S, \quad \forall i \quad \tilde{\mathbf{D}}_i \sim s\mathbf{D}_i .$$

Definition 6.2. Let the manifold \mathcal{H} of dual hexalinear tensors be defined as all $2 \times 2 \times 2 \times 2 \times 2 \times 2$ tensors H_{ijklmn} defined up to scale fulfilling $H_{ijklmn} = 0$ if not three of the indices are equal to 1 and the other three indices are equal to 2 and

$$\begin{aligned} H_{111222} &= -H_{222111}, & H_{112122} &= -H_{221211}, \\ H_{112212} &= -H_{221212}, & H_{112221} &= -H_{221112}, \\ H_{121122} &= -H_{212211}, & H_{121212} &= -H_{212121}, \\ H_{121221} &= -H_{212112}, & H_{122112} &= -H_{211221}, \\ H_{122121} &= -H_{211212}, & H_{122211} &= -H_{211122}, \\ -H_{121122} + H_{122211} + H_{112212} + H_{112221} &= 0, \\ H_{111222} - H_{112122} + H_{121212} - \\ H_{122121} - H_{122112} + H_{121221} &= 0, \\ H_{111222} + H_{112122} + H_{121122} + \\ H_{112212} + H_{121212} + H_{122112} &= 0, \\ H_{121122} + H_{122211} - H_{112212} + \\ 2H_{122121} + 2H_{121221} + H_{112221} &= 0, \\ H_{111222} + H_{112122} - H_{122211} - \\ H_{121212} - H_{122112} + H_{112221} &= 0 . \end{aligned} \quad (15)$$

Notice that 44 out of the 64 components are zero. There are 15 linear constraints among the 20 non-zero components. Since tensors are equal up to scale this can be considered as a linear projective space of dimension 4.

Theorem 6.2. The map (14) is well defined from \mathcal{U} to Q . For each tensor $Q \in Q$ there are two possible motions in \mathcal{U} .

This can be used to solve the structure and motion for six lines in four images in the following manner. First consider line directions only. Using the images estimate the dual hexalinear tensor linearly. From the tensor compute the two possible solutions for the 3D line directions. Use this to solve for the original problem. As in the (7 lines in 3 images) case only one of the two solutions survive to the original problem.

Another solution is to use the ordinary quadrifocal tensor under the assumption of affine cameras. Each line quadruple gives 9 constraints on the quadrifocal tensor. From 6 lines we get enough information to compute the tensor uniquely. From the tensor it is straightforward to calculate the camera matrices (see [8, 5]) and then the 3D structure of the lines using intersection.

Algorithm 6.1. (\geq six lines in four affine views).

1. Given at least four images of six points,
2. Calculate the dual hexalinear tensor fulfilling (15) and the hexalinear constraints

$$\sum_{i,j,k,l,m,n} H_{ijklmn} \mathbf{d}_1^i \mathbf{d}_2^j \mathbf{d}_3^k \mathbf{d}_4^l \mathbf{d}_5^m \mathbf{d}_6^n = 0 .$$

Notice that all constraints on the tensor are linear. It is thus straightforward to use a linear or svd type method.

3. Find the two solutions to structure from the tensor components.
4. For each solution to the structure calculate motion using resection.

7. Experiments

7.1. Six lines in three images

We have tried the minimal case of 6 lines in 3 images on real data. In Figure 1 is shown one of three images used. 6 lines of the sculpture was extracted and the 4 solutions to the structure and motion problem was computed as described above. In the figure is also shown the 3D structure of the lines for the 4 different solutions. Notice that only one of these four solutions (the lower left one) is correct. However, all of the four solutions have zero reprojection residual. More lines or images are needed in order to resolve the ambiguity.

7.2. Over determined cases

We have also tried the algorithms for the two over-determined cases (7 lines in 3 images, 6 lines in 4 images). These are similar in nature to the well-known eight point algorithm [12]. In order to obtain stable results one has to normalize the coordinates of the points and lines, cf. [7]. It is also useful to consider iterative improvements of the initial linear estimates, cf. [6], in which one tries to find the tensor that minimizes geometric errors while fulfilling the necessary constraints (4) or (10). After solving for the line structure one can perform a final bundle adjustment.

Notice that in both of these cases there are two solutions when considering line directions only. The relevant tensors are uniquely determined but each tensor correspond to two different structure and motion solutions for the line directions, cf. Theorem 4.2 and 6.2. However, only one of the two solutions survive when considering also the positions of the lines. For the original problem each tensor correspond to only one structure and motion solution.

For example in a simulated case of 7 lines in 3 views the $2 \times 2 \times 2$ tensor for the line directions is uniquely given by

$$T_{\bullet\bullet 1} = \begin{pmatrix} 0.18 & -0.48 \\ 0.017 & -0.49 \end{pmatrix}, \quad T_{\bullet\bullet 1} = \begin{pmatrix} 0.51 & 0.46 \\ 0.10 & 0.019 \end{pmatrix},$$

with two possible solutions for the upper left 2×3 block of the camera matrices. For the original problem the $3 \times 3 \times 3$ tensor is uniquely given by

$$T_{\bullet\bullet 1} = \begin{pmatrix} -0.51 & -0.46 & 0 \\ 0.18 & -0.48 & 0 \\ -66 & 48 & -0.52 \end{pmatrix},$$

$$T_{\bullet\bullet 2} = \begin{pmatrix} -0.10 & -0.019 & 0 \\ 0.017 & -0.49 & 0 \\ -13 & 1.2 & -0.10 \end{pmatrix},$$

$$T_{\bullet\bullet 3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0.019 & -0.39 & 0 \end{pmatrix},$$

with only one possible solution for the camera motion. Notice also that the 8 tensor components for the directions can be found rearranged in the first $2 \times 2 \times 2$ part of the tensor for the original problem.

Figure 2 illustrates this idea for the case of 4 views of 6 lines. A crude implementation of Algorithm 4 was used to find an initial estimate of structure and motion parameters. More work is needed to make our implementation stable. Nevertheless, the results are good enough to allow subsequent use of affine or projective bundle adjustment according to [16]. The final result has geometric errors in the order of 0.1 pixels.

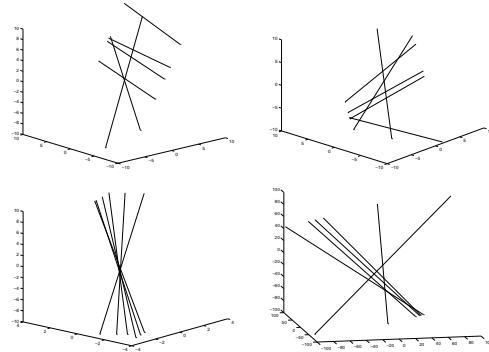
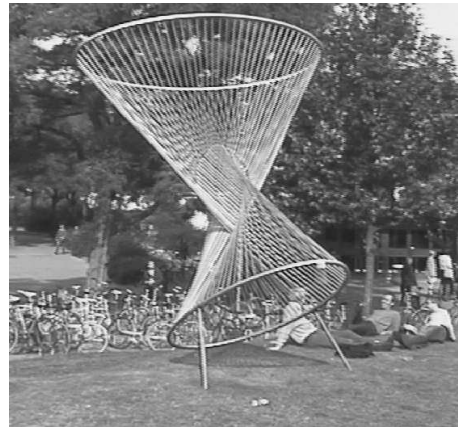


Figure 1. One image of three used in experiment. Four solutions to the structure and motion problem with 6 lines in 3 images using real image data. The lower left solution is the correct one.

8. Conclusions

In this paper, the minimal cases for structure and motion estimation of line features under the affine camera model, have been studied. It has been shown that the minimal cases are 5 lines in 4 images and 6 lines in 3 images. For these two minimal cases it has been shown that there are in general 4 solutions. In the process we have derived the necessary and sufficient conditions for trifocal tensors assuming affine cameras. We have also derived what we believe to be necessary and sufficient conditions for quadrifocal tensors assuming affine cameras.

The almost minimal cases of 7 lines in 3 images and 6 lines in 4 images have also been studied. It has been demonstrated that there are in general only one solution to these problems. We have also demonstrated how the so called dual hexalinear tensors can be used in the case of 4 images.

The algorithms have been studied both on simulated and real images. The 4 ambiguous solutions to one minimal case has been illustrated using real data. A further development would be to study the geometrical interpretation of

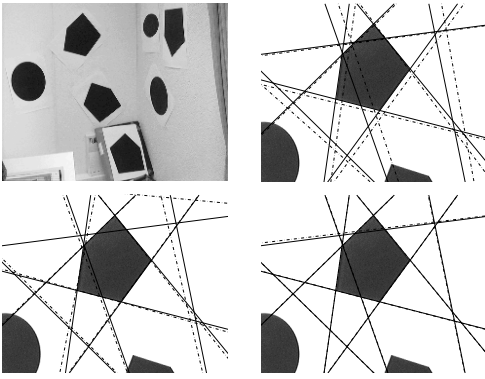


Figure 2. Top left: one of four images with extracted lines. Top right: reprojected and measured lines after linear solution (crude implementation). Bottom left: Result after bundle adjustment assuming affine cameras. Bottom right: Result after projective bundle adjustment.

these ambiguous solutions.

We hope that these new results can be used both for the theoretical understanding of multiple view geometry for lines under the affine camera model in general and for the understanding of these minimal cases in particular. The results have practical use both for obtaining initial estimates for structure and motion estimates but also for RANSAC algorithms for determining line correspondences.

References

- [1] S. Carlsson and D. Weinshall. Dual computation of projective shape and camera positions from multiple images. *Int. Journal of Computer Vision*, 27(3):227–241, 1998.
- [2] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Cambridge, Mass, 1993.
- [3] O. Faugeras and T. Papadopoulos. Grassman-cayley algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors. Technical Report 3225, Institut national de recherche en informatique et en automatique, july 1997.
- [4] O. Faugeras and T. Papadopoulos. A nonlinear method for estimating the projective geometry of three views. In *Proc. 6th Int. Conf. on Computer Vision, Mumbai, India*, pages 477–484, 1998.
- [5] R. Hartley. Computation of the quadrifocal tensor. In *Proc. 5th European Conf. on Computer Vision, Freiburg, Germany*, pages 20–35, 1998.
- [6] R. Hartley. Minimizing algebraic error in geometric estimation problems. In *Proc. 6th Int. Conf. on Computer Vision, Mumbai, India*, pages 469–476. IEEE Computer Society Press, 1998.
- [7] R. I. Hartley. In defence of the 8-point algorithm. In *Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, MA*, pages 1064–1070, 1995.
- [8] A. Heyden. A common framework for multiple-view tensors. In *Proc. 5th European Conf. on Computer Vision, Freiburg, Germany*, pages 3–19, 1998.
- [9] D. Jacobs. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 206–212, 1997.
- [10] F. Kahl and A. Heyden. Structure and motion from points, lines and conics with affine cameras. In *Proc. 5th European Conf. on Computer Vision, Freiburg, Germany*, pages 327–341, 1997.
- [11] F. Kahl and A. Heyden. Affine structure and motion from points, lines and conics. *Int. Journal of Computer Vision*, 1999. to appear.
- [12] C. Longuet-Higgins, H. computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, sept. 1981.
- [13] J. L. Mundy and A. Zisserman, editors. *Geometric Invariance in Computer Vision*. Artificial Intelligence series. MIT Press, 1992.
- [14] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(8):834–845, Aug. 1997.
- [15] L. Quan and Y. Ohta. A new linear method for euclidean motion/structure from three calibrated affine views. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 172–177, 1998.
- [16] K. Åström, F. Kahl, A. Heyden, and R. Berthilsson. A statistical approach to structure and motion from image features. In *Proc. International Workshop on Statistical Techniques in Pattern Recognition, Sydney*, volume 1451 of *Advances in Pattern Recognition, Lecture Notes in Computer Science*. Springer Verlag, 1998.
- [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 and A. Zisserman. Robust parameterization and computation of the trifocal tensor. In *British Machine Vision Conference*, 1996.
- [19] B. Triggs. Matching constraints and the joint image. In *Proc. ICCV’95*, pages 338–343. IEEE Computer Society Press, 1995.
- [20] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical report, INRIA, Sophia Antipolis, France, 1994.