

Minimal Projective Reconstruction for Combinations of Points and Lines in Three Views

Magnus Oskarsson¹, Andrew Zisserman² and Kalle Åström¹

¹Centre for Mathematical Sciences

Lund University, SE 221 00 Lund, SWEDEN

{magnuso, kalle}@maths.lth.se

²Department of Engineering Science

University of Oxford, Parks Road, Oxford OX1 3PJ, UK

az@robots.ox.ac.uk

Abstract

In this article we address the problem of projective reconstruction of structure and motion given only image data. In particular we investigate three novel minimal combinations of points and lines over three views, and give complete solutions and reconstruction methods for two of these cases: “four points and three lines in three views”, and “two points and six lines in three views”. We show that in general there are three and seven solutions respectively to these cases. The reconstruction methods are tested on real and simulated data. We also give tentative results for the case of nine lines in correspondence over three views, where experiments indicate that there may be up to 36 complex solutions.

1 Introduction

One of the core problems of computer vision is 3D reconstruction. Within the last years, reconstruction methods have been successfully extended to projective recon-

struction within an uncalibrated framework [9]. In this article we will investigate some minimal cases for projective reconstruction, where by a minimal case is meant that omission of some data gives an infinite number of solutions. We assume that only image data is given, and from this we seek projective solutions for the structure and the motion of the cameras.

Solving minimal cases to perform 3D reconstruction is not only of theoretical interest, it also is important in practice: solutions obtained from minimal cases can be used to bootstrap robust estimation algorithms such as RANSAC or LMS schema [4, 18, 25], and optimal estimation algorithms such as bundle adjustment [20]. In a RANSAC framework for solving structure and motion problems, a number of hypothetical matches are used to estimate a solution. Based on the rest of the available data this solution will then be accepted or rejected. In order to minimize the complexity of the procedure, one wants as small as possible sample size to estimate the initial solution. A minimal case is therefore a good way to get an initial estimate. A drawback of using minimal cases in RANSAC is that they can have multiple solutions. This will increase the complexity of the verification step.

One motivation for using lines to get initial estimates, is that they often can be measured with much higher accuracy than points can. This due to that a line can be estimated using a large number of extracted points in the image. The accuracy of the initial solutions will then benefit from this.

For three views and a projective reconstruction, the minimum number of points is 6 [10, 15], and the minimum number of lines is 9.

Linear algorithms have been developed for over-constrained solutions of at least 13 lines [6], and for combinations of lines and points [7]. Non-linear maximum likelihood estimators have also been developed for these over-constrained cases [18]. However, there has been little work on minimal cases for lines, or combinations of lines and points.

Over-constrained solutions have also been developed for other camera models for lines, and combinations of points and lines, including calibrated cameras [11, 17, 22, 23, 24], and affine cameras [1, 12, 16].

In the following, we will assume that the lines and points investigated are in general

positions. The results will not hold for critical configurations, which do exist. For lines cf. [2, 14], and for points, see [13]. We do not know of any work describing critical configurations for projective reconstruction of combinations of lines and points.

2 Problem formulation

A line in space has four degrees of freedom. A point in space has three degrees of freedom. In each image a point or a line gives two constraints on the unknown geometry. If we assume an uncalibrated pinhole camera then each camera has eleven degrees of freedom, cf. [9]. Since we work in a projective setting everything is defined up to a coordinate system with 15 degrees of freedom. A minimal projective structure and motion problem in m images given n points and k lines should hence fulfill:

$$(2n + 2k)m = 11m + 3n + 4k - 15. \quad (1)$$

In table 1 the different minimal cases for two and three cameras are summarised. There are other minimal problems for combinations of points and lines seen in four to nine cameras. The number of solutions are also given in the known cases. One can see in the table that as the size of the measured data-set increases, while holding the number of constraints equal to the number of estimated parameters, the complexity of the problem increases, with the result that a larger number of solutions occur. Our tentative results on the nine lines give indication of up to 36 solutions in this case.

In general adding information, i.e. adding one or more line or point, without increasing the number of images, or adding images without increasing the number of lines or points, will give an over-constrained problem. This will in general lead to a unique over-constrained solution. If we restrict ourselves to three images the minimal cases for combinations of points and lines are: “6 points”, “4 points and 3 lines”, “2 points and 6 lines” and “9 lines”. In this article we give solutions to the “4 points and 3 lines” problem as well as the “2 points and 6 lines” problem. We also give some tentative results on the case of nine lines.

Throughout the article, vectors are denoted in boldface and matrices in upper case boldface. Scalars are any plain letters or lower case Greek.

Points	Lines	Solutions
7	0	3

Two cameras

Points	Lines	Solutions
6	0	3
4	3	3
2	6	7
0	9	?

Three cameras

Table 1: The number of solutions for different minimal problems for two and three cameras.

We assume a perspective projection (uncalibrated pinhole camera) as the camera model. Thus the object space may be considered as embedded in $\mathbb{P}^3$ and the image space embedded in $\mathbb{P}^2$. The camera performs a projection from $\mathbb{P}^3$ to $\mathbb{P}^2$, and can be represented by a 3×4 matrix $\mathbf{P}_{3 \times 4}$ of rank 3 whose kernel is the projection centre. The relation between a point $\mathbf{X}$ in $\mathbb{P}^3$ and a point $\mathbf{x}$ in $\mathbb{P}^2$ can be written

$$\lambda \mathbf{x} = \mathbf{P}_{3 \times 4} \mathbf{X} . \quad (2)$$

An image line is represented by three homogeneous coordinates $\mathbf{l} = [l_x \ l_y \ l_z]^\top$ and the line is given by $\mathbf{l}^\top \mathbf{x} = 0$, where $\mathbf{x} = [u \ v \ w]^\top$ denotes points on the line. We will denote lines, points and cameras in view one without superscripts, in view two with primes and in view three with double primes. For example, a line in view two that is a projection of a line j in 3-space is denoted $\mathbf{l}'_j = [l'_{xj} \ l'_{yj} \ l'_{zj}]^\top$.

An important part of solving a minimal structure and motion problem is the choice of parameterization. A badly chosen parameterization will lead to a problem that is hard to solve and a good one may lead to the solution directly. In developing the solutions given in the following sections we have experimented with several different parameterizations and we only report the most tractable one for each case. Here we mention some of the issues involved in choosing the parameterization.

A basis in projective space has 15 degrees of freedom. Defining a basis with points in space is natural since five points have exactly $5 \times 3 = 15$ degrees of freedom. These points can be chosen in a canonical form, that has been used in many instances to

parameterize structure and motion problems, and it often leads to nice problem formulations.

Using lines and combinations of lines and points to parameterize the geometry is a bit more tricky as compared to just using points. There is no natural or canonical way to fix the coordinate system by specifying line coordinates. To fix a basis using lines will lead to the use of a maximum of three lines which have in total $3 \times 4 = 12$ degrees of freedom. The other degrees of freedom must be determined by the cameras or another point. In the case of fixing the degrees of freedom using the cameras this can be done in a number of different ways. The most intuitive is maybe putting the camera centres at specific points. In terms of the number of parameters that the cameras end up being parameterized by it is desirable to use as many points as are available. This is because a point gives just as many linear constraints on a camera as a line does, but it is determined by one parameter less than a line in space. This is the reason why problems involving many lines easily leads to polynomial equations of both high degree and with many unknown variables.

In order to be sure that a chosen parameterization is well defined, one has to verify that it defines a well defined homography so that any lines in general positions may be transferred to the given basis. For points and cameras this is straight-forward. For lines it is determined as follows: Each line will give four conditions on the homography. These linear conditions on the homography $\mathbf{H}$ can be written in the following way:

$$\mathbf{\Pi}_i^\top \mathbf{H} \mathbf{X}_j = 0, \quad i = 1, 2, \quad j = 1, 2, \quad (3)$$

where $\mathbf{H}$ transfers a line defined by the two points $(\mathbf{X}_1, \mathbf{X}_2)$ to a line defined by the intersection of the two planes $(\mathbf{\Pi}_1, \mathbf{\Pi}_2)$. Four lines are not enough to define a homography since there exists two invariants for four lines in $\mathbb{P}^3$, cf. [8, 5].

3 The case of four points and three lines

In this section we give an algorithm for solving the case of four points and three lines seen in three images. We will show that there is in general three solutions, of which some may be complex. The algorithm becomes linear given four points and four or

more lines. This in contrast to algorithms that only use the linear constraints on the trifocal tensor, which need at least four points and five lines. The methods are closely related to multipolynomial resultants, cf [3, 19].

3.1 Parameterization

As we are working within an uncalibrated projective setting, we may without restriction¹ introduce a projective coordinate system such that the 4 points in space are assigned to the canonical projective coordinates

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 & \mathbf{X}_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the first 4 image points in each image are assigned to

$$\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Using this choice of coordinates we get a special form of camera matrices. We have only used four points in our projective basis, which leaves the freedom to choose one more point. We fix the basis by letting the camera centre of camera one lie at the point $\mathbf{C}_1 = \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix}^\top$. This gives the following three cameras, parameterized with six parameters $(\alpha_1, \dots, \alpha_6)$:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \mathbf{P}' = \begin{bmatrix} \alpha_1 & 0 & 0 & 1 \\ 0 & \alpha_2 & 0 & 1 \\ 0 & 0 & \alpha_3 & 1 \end{bmatrix}, \mathbf{P}'' = \begin{bmatrix} \alpha_4 & 0 & 0 & 1 \\ 0 & \alpha_5 & 0 & 1 \\ 0 & 0 & \alpha_6 & 1 \end{bmatrix}.$$

3.2 Problem solution

We have used the four points to parameterize the cameras. We will now use the three lines to solve for the unknowns. The condition that three lines are images of a common

¹We are implicitly assuming that the 4 object points are projectively independent as well as the 4 points in each image.

3D line is

$$\text{rank } \mathbf{M}_j = \text{rank} \begin{bmatrix} \mathbf{P}^\top \mathbf{l}_j & \mathbf{P}'^\top \mathbf{l}'_j & \mathbf{P}''^\top \mathbf{l}''_j \end{bmatrix} = 2 .$$

In our case we have explicitly for each line j

$$\mathbf{M}_j = \begin{bmatrix} l_{xj} & \alpha_1 l'_{xj} & \alpha_4 l''_{xj} \\ l_{yj} & \alpha_2 l'_{yj} & \alpha_5 l''_{yj} \\ l_{zj} & \alpha_3 l'_{zj} & \alpha_6 l''_{zj} \\ S_j & S'_j & S''_j \end{bmatrix},$$

with

$$S_j = l_{xj} + l_{yj} + l_{zj}, \quad S'_j = l'_{xj} + l'_{yj} + l'_{zj}, \quad S''_j = l''_{xj} + l''_{yj} + l''_{zj}.$$

Expanding the four 3×3 -minors of $\mathbf{M}_j$ one gets the following four equations:

$$\mathbf{B}_{4 \times 12} \mathbf{Y} = \mathbf{0}_{4 \times 1}, \quad (4)$$

where

$$\mathbf{Y}^\top = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_2 \alpha_6 & \alpha_3 \alpha_5 & \alpha_1 \alpha_6 & \alpha_3 \alpha_4 & \alpha_1 \alpha_5 & \alpha_2 \alpha_4 & \alpha_4 & \alpha_5 & \alpha_6 \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} 0 & -l_{zj} l'_{yj} S''_j & l_{yj} l'_{zj} S''_j & l'_{yj} l''_{zj} S_j & -l'_{zj} l''_{yj} S_j & 0 \\ -l_{zj} l'_{xj} S''_j & 0 & l_{xj} l'_{zj} S''_j & 0 & 0 & l''_{zj} l'_{xj} S_j & \dots \\ -l'_{xj} l_{yj} S''_j & l_{xj} l'_{yj} S''_j & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & l_{xj} l'_{yj} l''_{zj} & -l_{xj} l'_{zj} l''_{yj} & -l_{yj} l'_{xj} l''_{zj} \\ \dots & 0 & 0 & 0 & l_{zj} l''_{yj} S'_j & -l''_{zj} l_{yj} S'_j \\ -l'_{zj} l''_{xj} S_j & 0 & 0 & l_{zj} l''_{xj} S'_j & 0 & -l_{xj} l''_{zj} S'_j \\ 0 & l'_{xj} l''_{yj} S_j & -l'_{yj} l''_{xj} S_j & l_{yj} l''_{xj} S'_j & -l_{xj} l''_{yj} S'_j & 0 \\ l_{yj} l'_{zj} l''_{xj} & l_{zj} l'_{xj} l''_{yj} & -l_{zj} l'_{yj} l''_{xj} & 0 & 0 & 0 \end{bmatrix}$$

Since $l_{xj} \mathbf{B}_1 - l_{yj} \mathbf{B}_2 + l_{zj} \mathbf{B}_3 = S_{1j} \mathbf{B}_1$ where $\mathbf{B}_i$ denotes the i 'th row of $\mathbf{B}$ we know that at most three of the four equations are linearly independent when considered as a system of equations in the unknown $(Y_1, \dots, Y_{12})$. We will only use the first three equations. These can be verified to be linearly independent (take for instance the sub-matrix of $\mathbf{B}$ consisting of rows one to three and columns two to four whose determinant

is clearly non-zero.) This results in 9 linear equations so we can express the first nine of the unknowns in $\mathbf{Y}$ in $(\alpha_4, \alpha_5, \alpha_6)$ by the Gauss-Jordan factorization $\tilde{\mathbf{B}}_{9 \times 12}$ of $\mathbf{B}_{9 \times 12}$. Inserting these expressions in the nonlinear internal constraints of $\mathbf{Y}$ gives a system of polynomial equations in $\{\alpha_4, \alpha_5, \alpha_6\}$ which can be written in the following way:

$$\mathbf{Q}_{6 \times 6} \begin{bmatrix} 1 & \alpha_4 & \alpha_5 & \alpha_4^2 & \alpha_5^2 & \alpha_4 \alpha_5 \end{bmatrix}^\top = \mathbf{0}_{6 \times 1}, \quad (5)$$

where $\mathbf{Q}_{6 \times 6}$ only depends on image data and α_6 . In order to have a solution to equation (5), $\mathbf{Q}$ has to be rank deficient and hence $\det \mathbf{Q} = 0$. Expanding the determinant of $\mathbf{Q}$ in α_6 gives a third degree polynomial in α_6 . Thus there are three solutions. In section 5.1 experiments on simulated data shows that there are cases with three distinct real solutions.

If we have four lines or more we can use equation (4) to get a linear solution. In this case we get for $n \geq 4$

$$\mathbf{B}_{3n \times 12} \mathbf{Y} = \mathbf{0}_{3n \times 1}. \quad (6)$$

In order to have a non-trivial solution the matrix $\mathbf{B}$ has to be rank deficient so the solution is given as the right null-space of $\mathbf{B}$. The scale of the solution is given by satisfying the nonlinear internal constraints inherited from $\mathbf{Y}$.

4 The case of two points and six lines

In this section we give a solution to the minimal case of two points and six lines viewed in three perspective views. We will show that in general there are seven solutions.

4.1 Parameterization

We will use the two points and two of the lines to parameterize the cameras. Let the two object points be given as $\mathbf{X}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\top$, $\mathbf{X}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top$, and the first two lines as $\mathbf{L}_1 = \begin{bmatrix} 0 & t & 1 & 1 \end{bmatrix}^\top$, $\mathbf{L}_2 = \begin{bmatrix} 1 & 1 & t & 1 \end{bmatrix}^\top$, $t \in \mathbb{R}$. One can not define a basis in $\mathbb{P}^2$ using two points and two lines, cf. [9], so the basis in the

images will be defined by three lines and one point. We let the two points project to

$$\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} u & 1 \\ v & 1 \\ w & 1 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{x}'_1 & \mathbf{x}'_2 \end{bmatrix} = \begin{bmatrix} u' & 1 \\ v' & 1 \\ w' & 1 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{x}''_1 & \mathbf{x}''_2 \end{bmatrix} = \begin{bmatrix} u'' & 1 \\ v'' & 1 \\ w'' & 1 \end{bmatrix},$$

and the first two lines to

$$\begin{bmatrix} \mathbf{l}_1 & \mathbf{l}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{l}'_1 & \mathbf{l}'_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{l}''_1 & \mathbf{l}''_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

This gives each camera three parameters, of the following form:

$$\mathbf{P} = \begin{bmatrix} u & 0 & \alpha_1 & -\alpha_1 \\ v & \alpha_1 - v & 0 & -\alpha_1 \\ w & \alpha_2 & \alpha_3 & -\alpha_1 \end{bmatrix}.$$

Similarly $\mathbf{P}'$ and $\mathbf{P}''$ are parameterized by $\alpha_4, \dots, \alpha_6$ and $\alpha_7, \dots, \alpha_9$. The scales of the camera matrices are inherited from $\mathbf{x}_1, \mathbf{x}'_1$ and $\mathbf{x}''_1$ respectively. We have fixed two lines and two points in space. This corresponds to $2 \cdot 4 + 2 \cdot 3 = 14$ degrees of freedom. This leaves one degree of freedom in the projective structure which can be fixed by letting $\alpha_1 = 1$.

4.2 Problem solution

The two points and two of the lines have been used in the parameterization. We will use the remaining four lines to solve the problem. We assume that we have made projective changes in the images such that $\mathbf{l}_3 = \mathbf{l}'_3 = \mathbf{l}''_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$. We will again use the fact that

$$\text{rank } \mathbf{M}_j = \text{rank} \begin{bmatrix} \mathbf{P}^\top \mathbf{l}_j & \mathbf{P}'^\top \mathbf{l}'_j & \mathbf{P}''^\top \mathbf{l}''_j \end{bmatrix} = 2, \quad j = 3 \dots 6.$$

We choose two equations for each line. These equations are obtained from $\mathbf{M}_j$ in the following way. One equation is given by taking the determinant of rows 1,3 and 4. The second equation is given by taking the determinant of rows 1,2 and 4. Since we are

only using two equations, a small number of spurious solutions are introduced (as will be shown later). These spurious solutions are, however, easy to identify later.

This will give rise to the following system of equations:

$$\mathbf{B}_{8 \times 16} \mathbf{Y} = \mathbf{0}_{8 \times 1}, \quad (7)$$

with

$$\mathbf{Y}_{16 \times 1}^\top = [\alpha_5 \ \alpha_6 \ \alpha_8 \ \alpha_9 \ \alpha_5\alpha_7 \ \alpha_6\alpha_7 \ \alpha_4\alpha_8 \ \alpha_4\alpha_9 \ \alpha_4\alpha_7 \ \alpha_2\alpha_4 \ \alpha_3\alpha_4 \ \alpha_2\alpha_7 \ \alpha_3\alpha_7 \ \alpha_4\alpha_7 \ 1]$$

and where $\mathbf{B}_{8 \times 16}$ depends on image data only.

We use the Gauss-Jordan factorization of $\mathbf{B}$ in (7):

$$\tilde{\mathbf{B}}_{8 \times 16} \mathbf{Y} = \begin{bmatrix} \mathbf{I}_{8 \times 8} & \mathbf{A}_{8 \times 8} \end{bmatrix} \mathbf{Y}^\top = \mathbf{0}_{8 \times 1}, \quad (8)$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{A}$ only depends on image data. We use the first four rows of $\tilde{\mathbf{B}}$ in (8) to express $(\alpha_5, \alpha_6, \alpha_8, \alpha_9)$ in terms of $(\alpha_2, \alpha_3, \alpha_4, \alpha_7)$, and substitute these expressions into the last four rows of (8). This gives four equations in $(\alpha_2, \alpha_3, \alpha_4, \alpha_7)$ which have the following structure:

$$\begin{cases} * \alpha_3 \alpha_4^2 + * \alpha_3 \alpha_4 + * \alpha_3 \alpha_4 \alpha_7 + * \alpha_3 \alpha_7 + * \alpha_4^2 + * \alpha_4^2 \alpha_7 + * \alpha_4 + * \alpha_4 \alpha_7 + * \alpha_7 = 0 \\ * \alpha_3 \alpha_7^2 + * \alpha_3 \alpha_4 + * \alpha_3 \alpha_4 \alpha_7 + * \alpha_3 \alpha_7 + * \alpha_7^2 + * \alpha_4 \alpha_7^2 + * \alpha_4 + * \alpha_4 \alpha_7 + * \alpha_7 = 0 \end{cases} \quad (9)$$

$$\begin{cases} * \alpha_2 \alpha_4^2 + * \alpha_2 \alpha_4 + * \alpha_2 \alpha_4 \alpha_7 + * \alpha_2 \alpha_7 + * \alpha_4^2 + * \alpha_4^2 \alpha_7 + * \alpha_4 + * \alpha_4 \alpha_7 + * \alpha_7 + * = 0 \\ * \alpha_2 \alpha_7^2 + * \alpha_2 \alpha_4 + * \alpha_2 \alpha_4 \alpha_7 + * \alpha_2 \alpha_7 + * \alpha_7^2 + * \alpha_4 \alpha_7^2 + * \alpha_4 + * \alpha_4 \alpha_7 + * \alpha_7 + * = 0 \end{cases} \quad (10)$$

Equation (9) is linear in α_3 and (10) is linear in α_2 , so we can use these equations to easily eliminate (α_2, α_3) . This leaves two polynomial equations in (α_4, α_7) of total degree five. Taking the resultant, cf. [3], of these two polynomials with respect to α_4 gives a polynomial of degree eleven in α_7 . Four of the eleven solutions are not true solutions, but arise from the fact that we can choose (α_4, α_7) so that row one and four in $\mathbf{M}_j$ are linearly dependent. This choice of (α_4, α_7) gives a solution to our chosen equations but will not lead to one for which all $\mathbf{M}_j$ have rank equal to two, and is therefore not true a solution.

This leaves seven solutions. As is shown in the experimental part there are indeed in some cases seven distinct real solutions to the problem.

Four points and three lines		
Nr of real sol.	1	3
Occurrences	10	30

Two points and six lines				
Nr of real sol.	1	3	5	7
Occurrences	1	8	6	6

Table 2: The number of real solutions for the two cases using simulated data.

5 Experiments

5.1 Results for real and simulated data

The methods for solving the two minimal cases described in sections 3 and 4 were implemented in Maple and tested on simulated data. The solutions give very small, close to machine precision, reprojection errors. In table 2 the number of real solutions for a number of runs on random data is shown. One can see that in some cases there are indeed 3 and 7 real solutions respectively for the two cases.

We have tried our algorithms on real data as well as the simulated. Figure 1 shows three images of a house complex. From these images we have manually extracted two corresponding points and six corresponding lines. The two points and two of the lines were used to make coordinate changes and parameterize the problem according to section 4.1. We then use the method described in section 4.2 to solve for the camera geometry. In this case three of the seven solutions were real.

Since it is a minimal problem the reprojection errors in the images are zero (except for numerical inaccuracies), but this is not an indication of how good the solutions are. To test our solutions eleven corresponding points were extracted in the three images. Using the computed cameras, the 3D structure was reconstructed linearly from the corresponding image points. The structure was then projected onto the images, using the computed cameras. For one of the three solutions the RMS error between measured and reprojected points was 2.099 pixels which was much smaller compared to the other two solutions. In figure 2 the original eleven points are shown with the reprojected points using this solution. Using a Newton-based optimization of the structure of the eleven points while holding the cameras fixed reduced the RMS-error to 2.043 pixels.

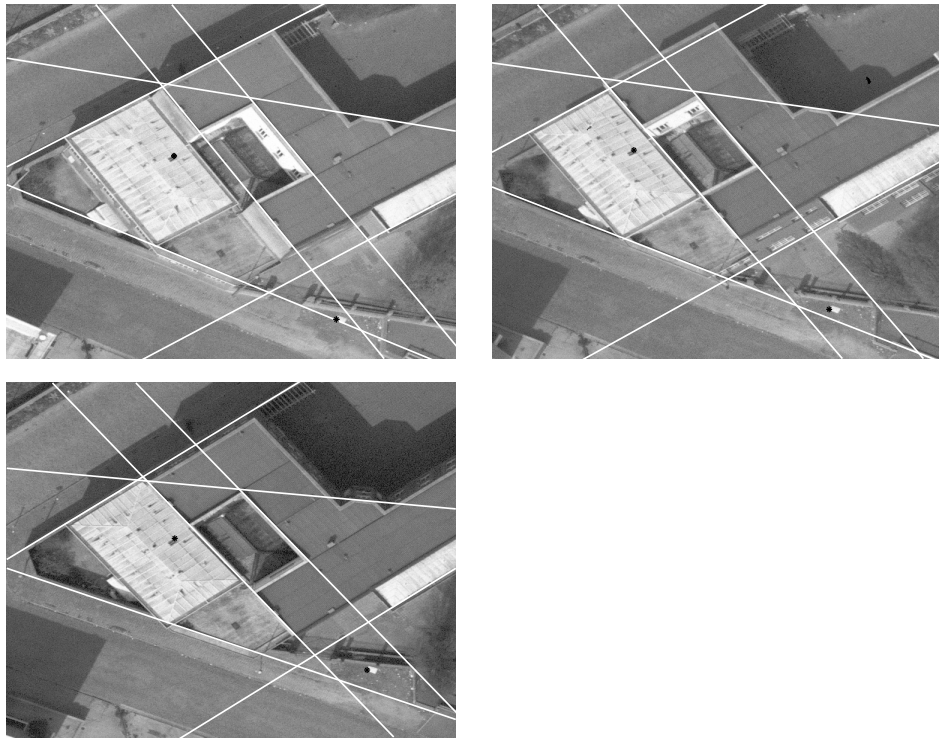


Figure 1: The three images used with six corresponding lines and two points.



Figure 2: Reprojection errors of the eleven points. The data points are marked with "*" and the reprojection points with "o".

After a full bundle adjustment, [20], the final solution had an RMS error equal to 0.4096 pixels.

We also applied the algorithm to another scene, shown in figure 3. The two dashed lines in combination with the two points shown in the figure were used in the parameterization. One should try to avoid choosing two lines that are close in direction in the parameterization, since this may lead to an unstable solution. Again in this case the image data gave rise to three real solutions. A number of additional lines, as well as the two conics in the sculpture, were extracted. The three solutions to the camera geometry were then used to compute 3D structure linearly from the image data. After reprojection, one of the solutions had much smaller reprojection errors than the two other solutions. The reprojection of this solution is shown in figure 4. The reprojection

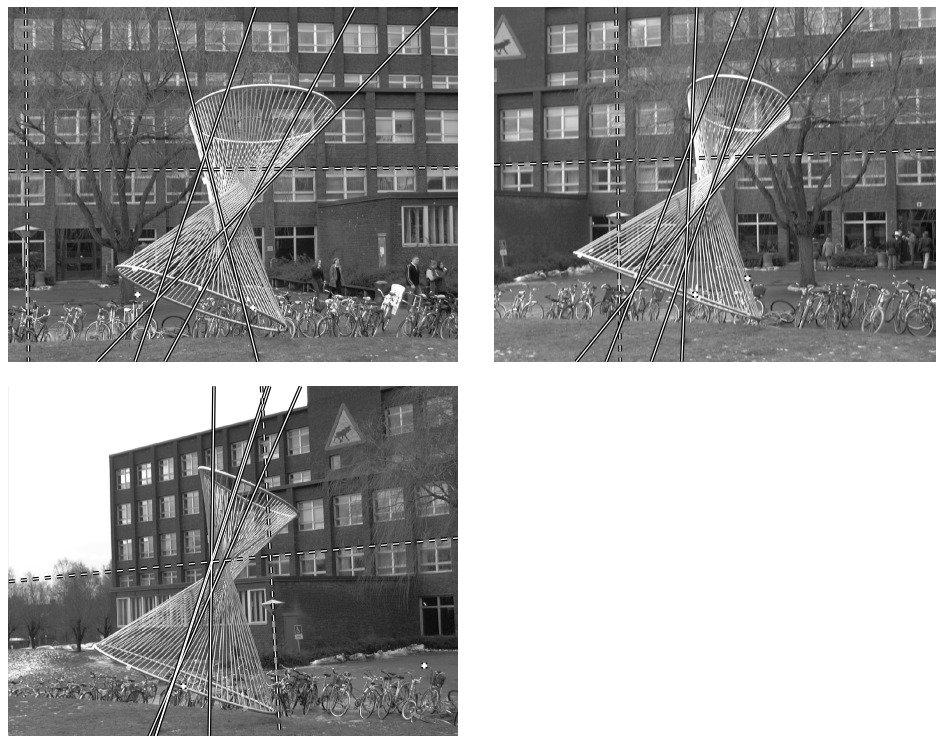


Figure 3: Three images used with the six lines and two points used in the reconstruction marked. The two dashed lines are the ones used to parameterize the problem.

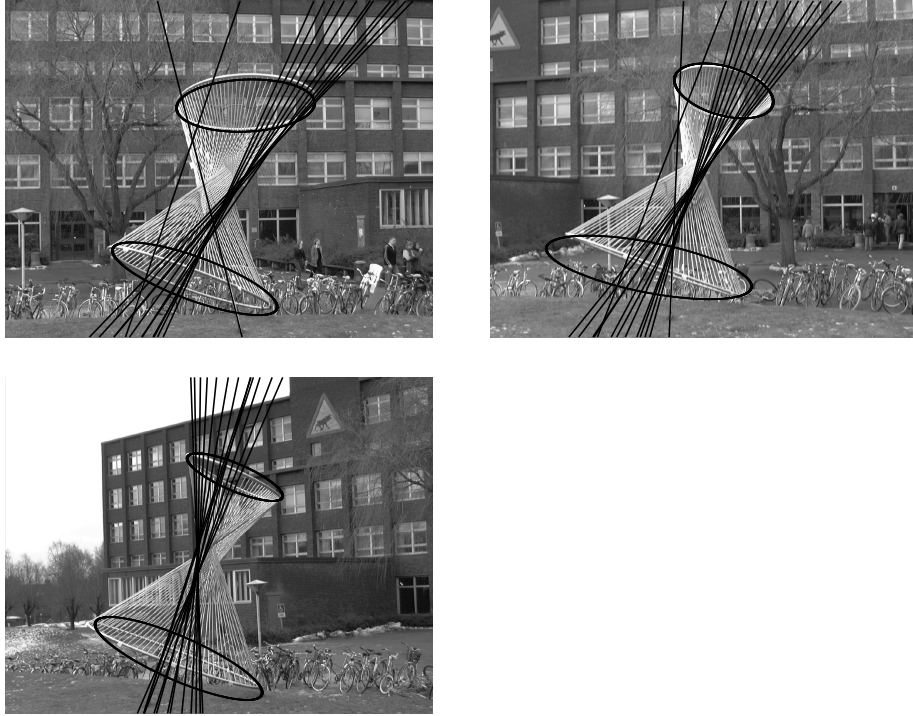


Figure 4: Reprojection of lines and conics – for the conics reasonable reprojection errors and for the lines the errors are very small.

errors for the lines are small. The errors for the conics are somewhat larger, especially the bottom one. A reason for this may be that the lines used in the estimation of camera geometry were extracted from the top part of the sculpture.

We also extracted 25 points from the top conic of the sculpture. The RMS errors for these points were 7.79 and 7.71 pixels for the linear and the optimized reconstruction respectively. After bundle adjustment the error decreased to 0.461 pixels. These experiments indicate that the obtained solutions can be used to bootstrap non-linear optimization methods or robust estimation schemes such as RANSAC.

5.2 Nine lines in three images

We will in this section give tentative results on the case of nine lines in correspondence over three views. We will parameterize the problem in a special way, and then use a numerical solver to solve this problem given some synthetic data.

Three of the lines are used in the parameterization. These lines are chosen as

$$\mathbf{L}_1 = \begin{bmatrix} 0 & t & 1 & 1 \end{bmatrix}^\top, \quad \mathbf{L}_2 = \begin{bmatrix} 1 & 0 & t & 1 \end{bmatrix}^\top, \quad \mathbf{L}_3 = \begin{bmatrix} t & 1 & 0 & 1 \end{bmatrix}^\top, \quad t \in \mathbb{R}.$$

The basis is then fixed by choosing the point $\mathbf{C}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top$ as the camera centre. A canonical bases in the images is chosen as described in section 4. This gives three cameras, parameterized with 12 parameters. The rank constraints then give 12 equations in the 12 unknowns. These can be chosen so that 6 equations are of total degree 3, and 6 others are of total degree 2.

Given a polynomial system an upper bound on the number of solutions is given by Bezout's theorem as the product of the total degrees of the polynomials. In our case we have 6 equations of total degree 3 and 6 others of total degree 2 which means a total of $2^6 \cdot 3^6 = 46656$ solutions. A better bound on the number of solutions is given by the so called **mixed volume** of the system, cf. [3]. The mixed volume looks at exactly which monomials that are present in each polynomial, as opposed to only looking at the largest total degree of each polynomial. If the polynomial is full, i.e. if all possible monomials up to the total degree have coefficients different from zero, then the mixed volume is equal to the bezout bound. However many polynomials have special structures and then the mixed volume gives a better bound on the number of solutions.

To solve the equations we have used a polynomial solver called PHC, which is described in [21]. This program starts with calculating the mixed volume of the system. In our case this turned out to be 413, which means that there are maximally 413 solutions to the nine lines problem. After the calculation of the mixed volume the solver proceeds by constructing a more easily solved system with the same structure as the original problem. It solves this system and the 413 solutions are propagated to the solution to the original system by a homotopy continuation method. Some of these solutions go out to infinity and are not solutions to the original problem. The PHC solver

Nr	1	2	3	4	5	6	7	8	9	10	11	12
PHC sol.	264	263	264	264	263	263	263	263	263	263	263	263
OK sol.	36	36	36	37	36	36	35	36	32	36	36	36
Real sol.	14	22	10	12	8	18	11	12	18	14	16	22

Table 3: The number of solutions in the nine lines case.

was used on a number of simulated nine lines cases. The resulting solutions were then verified numerically by inserting the solutions into the camera matrices and verifying the rank conditions by looking at the singular values. The spurious solutions that did not fulfill the rank constraints were removed, as well as those that lead to camera matrices with rank less than 3. The results on a number of runs is shown in table 3. In the table one can see that the number of solutions coming out from the PHC solver is quite stable. The verification is less stable since it depends on the threshold used to determine whether a singular value is zero or not. Since the estimated number of true solutions depends on this threshold it is hard to draw any conclusions on the exact number of solutions, even if we performed many more runs. Certainly in some cases there are errors. In run number 4 for instance, the number of complex solutions is indicated to be odd which clearly is not true. However these experiments give an indication that there are around 30-40 solutions to the nine lines case.

Each run of the PHC solver took around one hour on a SUN Ultra 5 running under Solaris. This means that this is not a viable method if one is interested in RANSAC for instance, where many samples may be required.

6 Conclusions

We have in this article investigated three novel minimal cases for projective reconstruction in three views.

We have given the number of solutions as well as algorithms for solving two of the cases. For the case of three lines and four points seen in three images there are up to three real solutions, and for the case of six lines and two points in three images there

are up to seven real solutions.

Both cases seem to perform reasonably well on real data. The algorithms give fast and accurate initial solutions, that can be used to bootstrap a RANSAC framework for structure and motion estimation.

The case of nine lines has been tentatively investigated numerically with solutions based on homotopy continuation methods. It is clear that there are no more than 413 solutions and the true number of solutions seems to be 36. However, more work is needed in order to solve this case.

One other minimal case in three views, that might be worth investigating in the future, is the case of three images of a quadric in combination with four points. This is related to the “four points and three lines” case in that three lines in general position define a ruled quadric.

References

- [1] K. Åström, A. Heyden, F. Kahl, and M. Oskarsson. Structure and motion from lines under affine projections. In *Proc. 7th Int. Conf. on Computer Vision, Kerkyra, Greece*, pages 285–292, 1999.
- [2] T. Buchanan. Critical sets for 3d reconstruction using lines. In G. Sandini, editor, *Proc. 2nd European Conf. on Computer Vision, Santa Margherita Ligure, Italy*, pages 730–738. Springer-Verlag, 1992.
- [3] D. Cox, J. Little, and D. O’Shea. *Using Algebraic Geometry*. Springer Verlag, 1998.
- [4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications-of-the-ACM*, 24(6):381–95, 1981.
- [5] I. Hartley, R. Lines and points in three views-a unified approach. In *Proceedings of the ARPA Image Understanding Workshop*, 1994.

- [6] R. Hartley. Projective reconstruction from line correspondences. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 903–907. IEEE Computer Society Press, 1994.
- [7] R. Hartley. Lines and points in three views and the trifocal tensor. *Int. Journal of Computer Vision*, 22(2):125–140, March 1997.
- [8] R. I. Hartley. Projective reconstruction and invariants from multiple images. *Pattern Analysis and Machine Intelligence*, 16(10):1036–1041, oct 1994.
- [9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [10] A. Heyden. *Geometry and Algebra of Multiple Projective Transformations*. PhD thesis, Lund Institute of Technology, Sweden, 1995.
- [11] R.J. Holt and A.N. Netravali. Uniqueness of solutions to structure and motion from combinations of point and line correspondences. *J. of visual communication and image representation*, 7(2):126–136, 1996.
- [12] R.J. Holt and A.N. Netravali. Motion and structure from line correspondences under orthographic projection. *Int. J. of Imaging Systems and Technology*, 8(3):301–312, 1997.
- [13] F. Kahl, R. Hartley, and K. Åström. Critical configurations for n-view projective reconstruction. In *Proc. Conf. Computer Vision and Pattern Recognition, Hawaii, USA*, 2001.
- [14] S. Maybank. The critical line congruences for reconstruction from three images. *Applicable Algebra in Engineering, Communication and Computing*, 6(2):89–113, 1995.
- [15] L. Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(1):34–46, 1995.

- [16] 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, August 1997.
- [17] M. E. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *Int. Journal of Computer Vision*, 4(3):171–183, 1990.
- [18] P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15(8):591–605, 1997.
- [19] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *Proc. 7th Int. Conf. on Computer Vision, Kerkyra, Greece*, pages 278–284. IEEE Computer Society Press, 1999.
- [20] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment: A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS. Springer Verlag, 2000.
- [21] J. Verschelde. Phcpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, 25(2):251–276, 1999.
- [22] T. Vieville and O. D. Faugeras. Feed-forward recovery of motion and structure from a sequence of 2d-lines matches. In *Proc. 3rd Int. Conf. on Computer Vision, Osaka, Japan*, pages 517–520, 1990.
- [23] J. Weng, T.S. Huang, and N. Ahuja. Motion and structure from line correspondences: Closed-form solution, uniqueness, and optimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(3), 1992.
- [24] J. Weng, Y. Liu, T.S. Huang, and N. Ahuja. Estimating motion/structure from line correspondences: A robust linear algorithm and uniqueness theorems. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 387–392, 1988.
- [25] Z. Zhang, R. Deriche, O. D. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.