

# Points, Lines and Planes and their Optimal Estimation

(slightly corrected version, 26.09.2001)

Stephan Heuel

Institut für Photogrammetrie, Universität Bonn  
Nussallee 15, D-53115 Bonn  
heuel@ipb.uni-bonn.de

**Abstract.** We present a method for estimating unknown geometric entities based on identical, incident, parallel or orthogonal observed entities. These entities can be points and lines in 2D and points, lines and planes in 3D. We don't need any approximate values for the unknowns. The entities are represented as homogeneous vectors or matrices, which leads to an easy formulation for a linear estimation model. Applications of the estimation method are manifold, ranging from 2D corner detection to 3D grouping.

## 1 Introduction

Constructing geometric entities such as points, lines and planes from given entities is a frequent task in Computer Vision. For example, in an image we may want to construct a junction point from a set of intersecting line segments (*corner detection*) or construct a line from a set of collinear points and line segments (*grouping*). In 3D, a space point can be reconstructed by a set of image points (*forward intersection*); as a last example, one may want to construct a polyhedral line given incident, parallel or orthogonal lines and planes (*3D-grouping*).

These geometric constructions can be described as an estimation task, where an unknown entity has to be fitted to a set of given observations in the least-squares sense. Unfortunately the underlying equations are nonlinear and quite difficult to handle in the Euclidean case. Because of the non-linearity, one needs approximate values for the unknowns, which are not always obvious to obtain.

This article presents a general model for estimating points, lines and planes without knowing approximate values. We propose to use algebraic projective geometry in 2D and 3D together with standard estimation methods. We first explore possible representations for geometric entities, obtaining a vector and a matrix for each entity. Then we express relations between the entities by simple, bilinear functions using the derived vectors and matrices. These relations can be directly used in a parameter estimation procedure, where the initial value can be obtained easily. We show possible applications and some test results validating the performance of the proposed method.

The proposed estimation model is based on (i) algebraic projective geometry, which has been extensively promoted in Computer Vision in the last decade (cf. [2],[6]) and (ii) integration of statistical information to the projective entities. Kanatani [7] presented a similar approach though he does not make full use of the elegant projective formulations, leading to complex expressions.

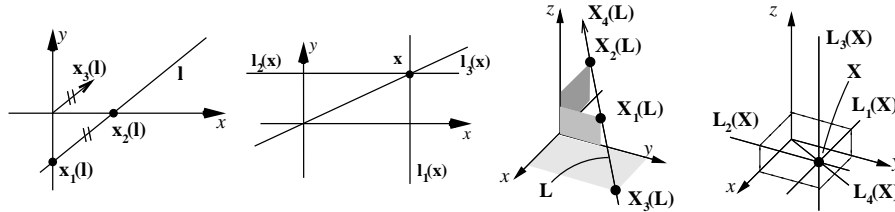
## 2 Representation of Geometric Entities

Assuming that both our observations and unknowns could be points, lines and planes in 2D and 3D, we first want to explore possible representations for these entities.

### 2.1 Vector Representation

Following the conventions in [4], points and lines in 2D are represented as homogeneous vectors  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{l}$ ,  $\mathbf{m}$ ; in 3D we have  $\mathbf{X}$ ,  $\mathbf{Y}$  for points,  $\mathbf{L}$ ,  $\mathbf{M}$  for lines and  $\mathbf{A}$ ,  $\mathbf{B}$  for planes. Euclidean vectors will be denoted with italic bold letters like  $\mathbf{x}$  for a Euclidean 2D point, cf. the first column in table 1. Furthermore we will use the canonical basis vectors  $\mathbf{e}_i$ ,  $\mathbf{E}_i = (0, \dots, \underset{i}{1}, \dots, 0)$  for points, lines or planes depending on the context.

For the geometric entities, each homogeneous vector contains a Euclidean part, indexed with a zero, and a homogeneous part, indexed with an  $h$ . This is chosen such that the distance of an element to the origin can be expressed as the norm of the Euclidean part divided by the norm of the homogeneous part, e.g. the distance of a plane to the origin is given by  $d_{A,0} = |A_0|/|\mathbf{A}_h|$ . The uncertainty of an entity can be described by the covariance matrix of this vector. Note that a line  $\mathbf{L} = (\mathbf{L}_h^\top, \mathbf{L}_0^\top)$  is a 6-vector which has to fulfill the Plücker condition  $\bar{\mathbf{L}}^\top \mathbf{L} = 0$  with the dual line  $\bar{\mathbf{L}} = (\mathbf{L}_0^\top, \mathbf{L}_h^\top)$ .



**Fig. 1.** (a) A line  $\mathbf{l}$  and the three common points  $\mathbf{x}_i(\mathbf{l})$  with the three lines  $\mathbf{e}_i$ . (b) A point  $\mathbf{x}$  and the three lines  $\mathbf{l}_i(\mathbf{x})$  incident to  $\mathbf{x}$  and the points  $\mathbf{e}_i$ . (c) A line  $\mathbf{L}$  and the four common points  $\mathbf{X}_i(\mathbf{L})$  with the four lines  $\mathbf{E}_i$ . (d) A point  $\mathbf{X}$  and the four lines  $\mathbf{L}_i(\mathbf{X})$  incident to  $\mathbf{X}$  and the points  $\mathbf{E}_i$ .

### 2.2 Matrix Representation

An entity is represented by a vector using its coordinates, but one can also use another representation: for example, a 2D line can be represented by its three homogeneous coordinates  $(a, b, c)$ , or implicitly by two points  $\mathbf{x}_1(\mathbf{l})$  and  $\mathbf{x}_2(\mathbf{l})$  on the  $x$ - resp.  $y$ -axis, see fig. 1(a). Additionally, one can choose the infinite point  $\mathbf{x}_3(\mathbf{l})$ , which lies in the direction of the line and is undefined, thus  $(0, 0, 0)^\top$ , if the line is at infinity. These three points can be constructed by  $\mathbf{x}_{1\dots 3}(\mathbf{l}) = \mathbf{l} \cap \mathbf{e}_{1\dots 3}$ . Writing  $\mathbf{x}_{1\dots 3}(\mathbf{l})$  in a  $3 \times 3$  matrix yields the skew-symmetric matrix  $\mathbf{S}(\mathbf{l})$  as defined in the second row of the table 1.

The same argument can be used to derive a skew-symmetric matrix  $\mathbf{S}(\mathbf{x})$ , which contains three lines  $\mathbf{l}_{1\dots 3}(\mathbf{x})$  incident to the point  $\mathbf{x}$ , see fig. 1(b). These

2D		vector	matrix
point	$\mathbf{x}^\top = (\mathbf{x}_0^\top, x_h) = (u, v; w)$	$\mathbf{S}(\mathbf{x}) = \begin{pmatrix} 0 & -w & v \\ w & 0 & -u \\ -v & u & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{l}_1(\mathbf{x})^\top \\ \mathbf{l}_2(\mathbf{x})^\top \\ \mathbf{l}_3(\mathbf{x})^\top \end{pmatrix} = \begin{pmatrix} (\mathbf{x} \wedge \mathbf{e}_1)^\top \\ (\mathbf{x} \wedge \mathbf{e}_2)^\top \\ (\mathbf{x} \wedge \mathbf{e}_3)^\top \end{pmatrix}$	
line	$\mathbf{l}^\top = (\mathbf{l}_h^\top, l_0) = (a, b; c)$	$\mathbf{S}(\mathbf{l}) = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1(\mathbf{l})^\top \\ \mathbf{x}_2(\mathbf{l})^\top \\ \mathbf{x}_3(\mathbf{l})^\top \end{pmatrix} = \begin{pmatrix} (\mathbf{l} \cap \mathbf{e}_1)^\top \\ (\mathbf{l} \cap \mathbf{e}_2)^\top \\ (\mathbf{l} \cap \mathbf{e}_3)^\top \end{pmatrix}$	
3D		vector	matrix
point	$\mathbf{X}^\top = (\mathbf{X}_0^\top, X_h) = (U, V, W; T)$	$\overline{\Pi}^\top(\mathbf{X}) = \begin{pmatrix} X_h I & \mathbf{S}(\mathbf{X}_0) \\ -\mathbf{X}_0^\top & \mathbf{0}^\top \end{pmatrix} = \begin{pmatrix} T & 0 & 0 & 0 & W & -V \\ 0 & T & 0 & -W & 0 & U \\ 0 & 0 & T & V & -U & 0 \\ -U & -V & -W & 0 & 0 & 0 \end{pmatrix}$	
line	$\mathbf{L}^\top = (\mathbf{L}_h^\top, \mathbf{L}_0^\top) = (L_1, L_2, L_3; L_4, L_5, L_6)$	$\Gamma(\mathbf{L}) = \begin{pmatrix} -\mathbf{S}(\mathbf{L}_h) & -\mathbf{L}_0 \\ \mathbf{L}_0^\top & 0 \end{pmatrix} = \begin{pmatrix} 0 & L_3 & -L_2 & -L_4 \\ -L_3 & 0 & L_1 & -L_5 \\ L_2 & -L_1 & 0 & -L_6 \\ L_4 & L_5 & L_6 & 0 \end{pmatrix}$ $\overline{\Gamma}(\mathbf{L}) = \Gamma(\overline{\mathbf{L}}) = \begin{pmatrix} -\mathbf{S}(\mathbf{L}_0) & -\mathbf{L}_h \\ \mathbf{L}_h^\top & 0 \end{pmatrix} = \begin{pmatrix} 0 & L_6 & -L_5 & -L_1 \\ -L_6 & 0 & L_4 & -L_2 \\ L_5 & -L_4 & 0 & -L_3 \\ L_1 & L_2 & L_3 & 0 \end{pmatrix}$	
plane	$\mathbf{A}^\top = (\mathbf{A}_h^\top, A_0) = (A, B, C; D)$	$\overline{\overline{\Pi}}^\top(\mathbf{A}) = \begin{pmatrix} -\mathbf{S}(\mathbf{A}_h) & A_0 I \\ -\mathbf{A}_h^\top & \mathbf{0}^\top \end{pmatrix} = \begin{pmatrix} 0 & C & -B & D & 0 & 0 \\ -C & 0 & A & 0 & D & 0 \\ B & -A & 0 & 0 & 0 & D \\ 0 & 0 & 0 & -A & -B & -C \end{pmatrix}$	

**Table 1.** Points, lines and planes in 2D and 3D represented as homogeneous vectors  $\mathbf{x}$ ,  $\mathbf{l}$  and matrices  $\mathbf{S}(\mathbf{x})$ ,  $\mathbf{S}(\mathbf{l})$ , resp in 3D represented as homogeneous vectors  $\mathbf{X}$ ,  $\mathbf{L}$ ,  $\mathbf{A}$  and matrices  $\overline{\Pi}^\top(\mathbf{X})$ ,  $\Gamma(\mathbf{L})$ ,  $\overline{\overline{\Pi}}^\top(\mathbf{A})$ . Each row of a homogeneous matrix represents a vector of the other entity. All three row entities of a matrix in turn define the given entity, see text for details.

matrices have rank 2, because only two rows in the matrix  $\mathbf{S}(\bullet)$  are needed to construct the third one.

In a similar manner we can define matrices for points, lines and planes in 3D: for example, a point  $\mathbf{X}$  can be defined by a set of four intersecting lines  $\mathbf{L}_{1\dots 4}(\mathbf{X})$ , where three lines are each parallel to one axis and the fourth line intersects the origin, see fig. 1(d). Since a line is represented by a 6-vector, we obtain a  $4 \times 6$  matrix  $\overline{\Pi}^\top(\mathbf{X}) = (\mathbf{L}_1(\mathbf{X}) \dots \mathbf{L}_4(\mathbf{X}))^\top$  as a representation of a point  $\mathbf{X}$ . Note that the six columns of  $\overline{\Pi}^\top(\mathbf{X})$  can be interpreted as six planes, which are incident to the point  $\mathbf{X}$  and to the six space lines  $\overline{\mathbf{E}}_i$ .

A space line can be determined by four points  $\mathbf{X}_{1\dots 4}(\mathbf{L})$ , three of them are on the axis-planes  $\mathbf{E}_{1\dots 3}$ , the fourth one lies on the infinite plane  $\mathbf{E}_4$  in direction of the line, see fig. 1(c). These four points define  $4 \times 4$  skew-symmetric matrix  $\overline{\Gamma}(\mathbf{L})$ . Only two rows of the matrix  $\overline{\Gamma}(\mathbf{L})$  are linearly independent, the selection of the rows depend on the position and direction of the line. As a space line can also be represented by four planes  $\mathbf{A}_{1\dots 4}(\mathbf{L})$ , we can define a line matrix based on these for planes, yielding  $\Gamma(\mathbf{L})$ , cf. table 1. In the literature,  $\Gamma(\mathbf{L})$  and  $\overline{\Gamma}(\mathbf{L})$  are referred as Plücker matrices, e.g. in [6].

The matrix of a plane can be defined dually to the matrix of a point by  $\overline{\overline{\Pi}}^\top(\mathbf{A})$ , as in table 1, right column, last row. Note that  $\mathbf{S}(\mathbf{x})$ ,  $\mathbf{S}(\mathbf{l})$ ,  $\overline{\Pi}^\top(\mathbf{X})$ ,  $\Gamma(\mathbf{L})$ ,  $\overline{\overline{\Pi}}^\top(\mathbf{A})$  are not only representations of points, lines and planes, but also the Jacobians for constructing elements by join  $\wedge$  and intersection  $\cap$ , see [4].

### 3 Relations between Unknown and Observed Entities

When estimating an unknown entity, all the observations are related to the unknown in some way. In this work we assume the relations to be either *incidence*, *equality*, *parallelity* and *orthogonality*; for example an unknown space line  $\mathbf{M}$  may have a list of incident points  $\mathbf{X}_i$ , incident lines  $\mathbf{L}_i$  and parallel planes  $\mathbf{A}_i$  as observed entities. In this section we want to derive algebraic expressions for a relation between two entities.

First we want to focus on incidence and equality relations in 3D: the simplest incidence relation is a point-plane incidence, since it is only the dot product of the two vectors:  $\mathbf{X}^\top \mathbf{A} = 0$ . Furthermore, the incident relation of two lines can be expressed by a dot product:  $\overline{\mathbf{M}}^\top \mathbf{L} = 0$ . From  $\mathbf{X}^\top \mathbf{A}$  and  $\overline{\mathbf{M}}^\top \mathbf{L}$  we can derive all other possible incidence and equality relations: for example, if two lines  $\mathbf{L}$  and  $\mathbf{M}$  are equal, each of the four points  $\mathbf{X}_{1\dots 4}(\mathbf{M})$  must be incident to each of the four planes  $\mathbf{A}_{1\dots 4}(\mathbf{L})$ . Therefore the matrix product of  $\overline{\Gamma}(\mathbf{L}) \Gamma(\mathbf{M}) = \mathbf{0}$  must be zero. Another example is a line  $\mathbf{L}$  which is incident to plane  $\mathbf{A}$  if the four lines  $\mathbf{L}_{1\dots 4}(\mathbf{A})$  are incident to the line  $\mathbf{L}$ . All possible incidence and equality relations between unknowns and observations are listed in table 3.

obs.→	$\mathbf{x}_i$		$\mathbf{l}_i$	
↓ unk.	relation	constraint	relation	constraint
$\mathbf{y}$	$\mathbf{x}_i \equiv \mathbf{y}$	$\mathbf{S}(\mathbf{x}_i) \mathbf{y} = \mathbf{0}$	$\mathbf{l}_i \ni \mathbf{x}_i$	$\mathbf{l}_i^\top \mathbf{y} = 0$
$\mathbf{m}$	$\mathbf{x}_i \ni \mathbf{m}$	$\mathbf{x}_i^\top \mathbf{m} = 0$	$\mathbf{l}_i \equiv \mathbf{m}$	$\mathbf{S}(\mathbf{l}_i) \mathbf{m} = \mathbf{0}$
			$\mathbf{l}_i \parallel \mathbf{m}$	$(\mathbf{l}_i^\perp) \mathbf{m} = 0$
			$\mathbf{l}_i \perp \mathbf{m}$	$\mathbf{l}_i^t \mathbf{m} = 0$

**Table 2.** Possible incidence, equality, parallelity and orthogonality relations between lists of observed points  $\mathbf{x}_i$  and of lines  $\mathbf{l}_i$  and unknown points  $\mathbf{y}$  and lines  $\mathbf{m}$ .

o.→	$\mathbf{X}_i$		$\mathbf{L}_i$		$\mathbf{A}_i$	
↓ u.	relation	constraint	relation	constraint	relation	constraint
$\mathbf{Y}$	$\mathbf{X}_i \equiv \mathbf{Y}$	$\Pi(\mathbf{X}_i) \mathbf{Y} = \mathbf{0}$	$\mathbf{L}_i \ni \mathbf{Y}$	$\Gamma(\mathbf{L}_i) \mathbf{Y} = \mathbf{0}$	$\mathbf{A}_i \ni \mathbf{Y}$	$\mathbf{A}_i^\top \mathbf{Y} = 0$
$\mathbf{M}$	$\mathbf{X}_i \in \mathbf{M}$	$\overline{\Pi}^\top(\mathbf{X}_i) \mathbf{M} = \mathbf{0}$	$\mathbf{L}_i \equiv \mathbf{M}$	$\overline{\Gamma}(\mathbf{L}_i) \Gamma(\mathbf{M}) = \mathbf{0}$	$\mathbf{A}_i \ni \mathbf{M}$	$\Pi^\top(\mathbf{A}_i) \mathbf{M} = 0$
			$\mathbf{L}_i \cap \mathbf{M}_i \neq \emptyset$	$\overline{\mathbf{L}}_i^\top \mathbf{M} = 0$		
$\mathbf{B}$	$\mathbf{X}_i \in \mathbf{B}$	$\mathbf{X}_i^\top \mathbf{B} = 0$	$\mathbf{L}_i \in \mathbf{B}$	$-\overline{\Gamma}(\mathbf{L}_i) \mathbf{B} = \mathbf{0}$	$\mathbf{A}_i \equiv \mathbf{B}$	$\Pi(\mathbf{A}_i) \mathbf{B} = \mathbf{0}$

**Table 3.** Possible incidence and equality relations  $\in, \equiv$  between lists of observed space points  $\mathbf{X}_i$ , lines  $\mathbf{L}_i$ , and planes  $\mathbf{A}_i$  and unknown space points  $\mathbf{Y}_i$ , lines  $\mathbf{M}_i$  and planes  $\mathbf{B}_i$  and their corresponding algebraic constraint.

Parallelity and orthogonality relations refer to the homogeneous parts of the vectors: in 3D we have to test whether the (Euclidean) direction  $\mathbf{L}_h$  of the line  $\mathbf{L}$  is perpendicular resp. parallel to the plane normal  $\mathbf{A}_h$ . In general, these relations can be expressed by the dot product or by the cross product of the two vectors, where the cross product can be written using the skew-symmetric matrix  $\mathbf{S}$ ; the relations are listed in table 4.

Two things are important to note: (i) all algebraic constraints for the relations and 2D and 3D (for 2D, see table 2) are bilinear with respect to each entity, cf. [4]. (ii) We do not have to compute every dot product relevant for a specific relation. It is sufficient to select as many as the degree of freedoms of the relation: for example, the incidence of a point  $\mathbf{X}$  and a line  $\mathbf{L}$  has two degrees of freedom, therefore we can select two rows  $\mathbf{A}_{j,k}^\top(\mathbf{L})$  out of  $\Gamma(\mathbf{L})$  and compute the dot product  $\mathbf{A}_j^\top(\mathbf{L}) \mathbf{X} = 0$  and  $\mathbf{A}_k^\top(\mathbf{L}) \mathbf{X} = 0$ . The selection depends on the coordinates  $L_j$  of line  $\mathbf{L}$ , it is numerically safe to take the two rows with the largest  $|L_{j,k}|$ .

obs. → ↓ unk.	$\mathbf{L}_i$		$\mathbf{A}_i$	
	relation	constraint	relation	constraint
$\mathbf{M}$	$\mathbf{L}_i \parallel \mathbf{M}$	$S(\mathbf{L}_{i_h}) \mathbf{M}_h = \mathbf{0}$	$\mathbf{A}_i \parallel \mathbf{M}$	$\mathbf{A}_{i_h}^\top \mathbf{M}_h = 0$
	$\mathbf{L}_i \perp \mathbf{M}$	$\mathbf{L}_{i_h}^\top \mathbf{M}_h = 0$	$\mathbf{A}_i \perp \mathbf{M}$	$S(\mathbf{A}_{i_h}) \mathbf{M}_h = \mathbf{0}$
$\mathbf{B}$	$\mathbf{L}_i \parallel \mathbf{B}$	$\mathbf{L}_{i_h}^\top \mathbf{B}_h = 0$	$\mathbf{A}_i \parallel \mathbf{B}$	$S(\mathbf{A}_{i_h}) \mathbf{B}_h = \mathbf{0}$
	$\mathbf{L}_i \perp \mathbf{B}$	$S(\mathbf{L}_{i_h}) \mathbf{B}_h = \mathbf{0}$	$\mathbf{A}_i \perp \mathbf{B}$	$\mathbf{A}_{i_h}^\top \mathbf{B}_h = 0$

**Table 4.** Possible orthogonality and parallelity relations  $\perp, \parallel$  between observed list of lines  $\mathbf{L}_i$  and list of planes  $\mathbf{A}_i$  and unknown lines  $\mathbf{M}$  and  $\mathbf{A}$ . The lower index  $h$  indicates the use of the homogeneous part of an entity, cf. table 1.

#### 4 Estimation of an Unknown Entity

We now have the relations between an unknown entity  $\beta$  and a list of observation entities  $\gamma_i$  expressed as an implicit form  $\mathbf{g}_i(\beta; \gamma_i) = \mathbf{0}$ .

Taking this equation, we can use the Gauss-Helmert model for estimating the unknown  $\beta$ , cf. [8] or [9]. Since the Gauss-Helmert model is a linear estimation model, we need the Jacobians  $\partial \mathbf{g}_i(\beta; \gamma_i) / \partial \gamma_i$  and  $\partial \mathbf{g}_i(\beta; \gamma_i) / \partial \beta$ . We already have given them in sec. 3 because all our expressions were bilinear: in the tables 2,3 and 4 the Jacobians  $\partial \mathbf{g}_i(\beta; \gamma_i) / \partial \beta$  are given for each combination of unknown  $\beta$  and observation  $\gamma_i$ <sup>1</sup>.

As the estimation is iterative, we need an initial value for the unknown  $\beta$ . Since we do not need the observations, we can minimize the algebraic distance  $\Omega$  (cf. [1], p. 332 & 377):

$$\Omega = \sum_i \mathbf{g}_i^\top \mathbf{g}_i = \beta^\top \left( \sum_i \mathbf{A}_i \mathbf{A}_i^\top \right) \beta \quad \text{with } \mathbf{A}_i = \frac{\partial \mathbf{g}_i(\beta; \gamma_i)}{\partial \beta}$$

This leads to an eigenvector problem for the matrix  $\sum_i \mathbf{A}_i \mathbf{A}_i$ : the smallest normalized eigenvector gives the approximate value for the unknown  $\beta$ .

In section 3 we have seen that the Jacobians  $\mathbf{A}_i$  may not have full rank, which leads to a singular covariance matrix  $\Sigma_{gg}$  inside the Gauss-Helmert model. One can either compute the pseudoinverse as in [4], or select the rows of the Jacobians, so that the Jacobian has the same rank number as the number of degrees of freedom for the relation.

<sup>1</sup> for equality relations, one can also use a different Jacobian as proposed in [4]

## 5 Example Tasks

We have implemented the proposed estimation as a unique generic algorithm with all 2D and 3D entities and all proposed onstraints. The input of the algorithm is the type of the unknown and an arbitrary set of observations . We will denote observations with their corresponding constraint types  $\in, \equiv, \parallel, \perp$  as upper index; e.g. if an unknown point  $\mathbf{y}$  should lie on a set of lines observation, we have  $\mathbf{g}(\mathbf{y}; \mathbf{l}_i^\ni) = 0 \Leftrightarrow \mathbf{l}_i^\ni \ni \mathbf{y}$ , denoting the observed lines  $\mathbf{l}_i^\ni$  with upper index  $\ni$ , since they are incident to the point  $\mathbf{y}$ .

Using the same estimation algorithm, we can e.g. solve the following tasks among many others (the third and the fourth task will be evaluated in sec. 6):

- *Corner detection in 2D.* We have an unknown point  $\mathbf{y}$  and a set of observed lines  $\mathbf{l}_i^\ni$  for which the unknown is incident,  $\mathbf{l}_i^\ni \ni \mathbf{y}$ . The implicit functional model is  $\mathbf{g}(\mathbf{y}; \mathbf{l}_i^\ni) = \mathbf{0}$ , the Jacobians are  $\mathbf{S}(\mathbf{y})$  and  $\mathbf{S}(\mathbf{l}_i^\ni)$ .
- *Grouping in 2D.* We have an unknown line  $\mathbf{m}$  and the following lists of observations: (1) incident points  $\mathbf{x}_i^\in$ , (2) collinear (i.e. equal) lines  $\mathbf{l}_i^\equiv$ , (3) parallel lines  $\mathbf{l}_i^\parallel$ , (4) orthogonal lines  $\mathbf{l}_i^\perp$ . The implicit functional model is  $\mathbf{g}(\mathbf{y}; \mathbf{x}_i^\in, \mathbf{l}_i^\equiv, \mathbf{l}_i^\parallel, \mathbf{l}_i^\perp) = \mathbf{0}$ . The Jacobians for  $\partial\mathbf{g}(\mathbf{m}; \gamma_i)/\partial\mathbf{m}$  are listed in table 2.
- *Forward intersection.* We want to construct a 3D point  $\mathbf{Y}$  out of a set of points  $\mathbf{x}_j$  and lines  $\mathbf{l}_{i,j}$  in  $n$  images  $j \in \{1 \dots n\}$ . Inverting the projections  $\mathbf{P}_j$ , we can obtain 3D lines  $\mathbf{L}(\mathbf{x}_j)$  and planes  $\mathbf{A}(\mathbf{l}_{i,j})$ , see [2]. and [3]. We can now estimate the unknown 3D point  $\mathbf{M}$ : the implicit functional model is  $\mathbf{g}(\mathbf{Y}; \mathbf{L}(\mathbf{x}_j), \mathbf{A}(\mathbf{l}_{i,j})) = \mathbf{0}$ . The Jacobians used here are  $\Gamma(\mathbf{L}(\mathbf{x}_j))$ ,  $\overline{\Pi}^\top(\mathbf{Y})$ ,  $\mathbf{A}(\mathbf{l}_{i,j})^\top$  and  $\mathbf{Y}^\top$ . Note that we can also construct 3D lines from sets of 2D points and lines by forward intersection, see also sec. 6.
- *Grouping in 3D.* A 3D line representing one edge of a cube is unknown, the observed entities are: (1) incident points  $\mathbf{X}_i^\in$ , (2) incident lines  $\mathbf{L}_i^\cap$ , (3) collinear (i.e. equal) lines  $\mathbf{L}_i^\equiv$ , (4) parallel lines  $\mathbf{L}_i^\parallel$ , (5) orthogonal lines  $\mathbf{L}_i^\perp$ , (6) incident planes  $\mathbf{A}_i^\ni$ , (7) parallel planes  $\mathbf{A}_i^\parallel$  and (8) orthogonal planes  $\mathbf{A}_i^\perp$ . All observed entities can be derived by the other 9 edges, 8 corners and 6 surfaces of the cube. We have  $\mathbf{g}(\mathbf{M}; \mathbf{X}_i^\in, \mathbf{L}_i^\cap, \mathbf{L}_i^\equiv, \mathbf{L}_i^\parallel, \mathbf{L}_i^\perp, \mathbf{A}_i^\ni, \mathbf{A}_i^\parallel, \mathbf{A}_i^\perp) = \mathbf{0}$  and the Jacobians listed in table 3 and 4. In the same manner we can also construct a point representing a cube corner or a plane representing a cube surface.

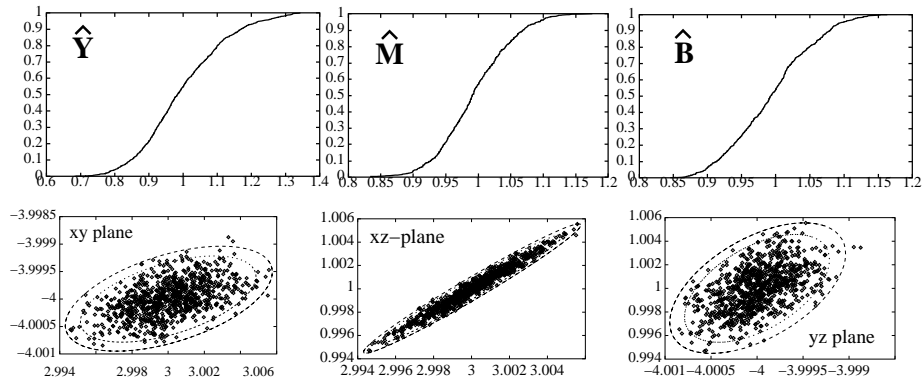
## 6 Experimental Results

*Artificial Data.* To validate our algorithm, we have tested it for the previously described 3D grouping task in section 5 using artificial data and applying all proposed constraints. Artificial data provides ground-truth which is necessary for validating an estimation method. We estimated a 3D point, line and plane being part of translated and rotated cubes of size 1. For each of the 6 corners, 12 edges and 6 surfaces of the cube, 5 observations have been generated randomly with a Gaussian error of 0.1% and 1%. When taking into account all possible constraints, the redundancy of the estimations are 57 for the unknown point,

176 for the unknown line and 162 for the unknown plane. We drew 1000 sample cubes and then estimated 1000 points, lines and planes. In fig. 2, the first row shows the cumulative histogram for the estimated  $\hat{\sigma}_0^2$  for the points  $\hat{\mathbf{Y}}$ ,  $\hat{\mathbf{M}}$  and planes  $\hat{\mathbf{B}}$ , over all estimates we get an average of  $\overline{\hat{\sigma}_0^2} = 0.988$ . The second row depicts the scatter diagram for the estimated point  $\hat{\mathbf{Y}}$ , projected onto the three principal planes. Furthermore, the  $\chi^2$ -test on bias was not rejected.

For the given datasets, all estimations converged within 3 to 4 iterations. The stopping criteria for the iteration was defined as follows: the corrections to the values of  $\beta$  should be less than 1% with respect to its standard deviation. The algorithm was implemented in the scripting language Perl and takes about 1 [sec] for 50 point resp. plane observations with a factor 1.5 slower for lines.

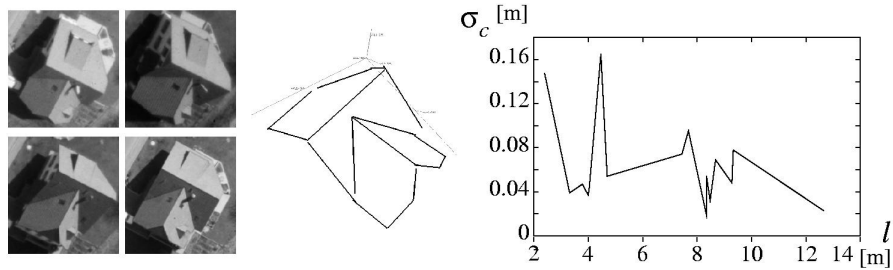
*Real Data.* We have tested the task of forward intersection of 3D lines using corresponding points and lines from four overlapping aerial images. Interactively we specified sets of 2D points and line segments which correspond to the same 3D segments. The 2D features were automatically extracted by a polymorphic point-feature extraction method called FEX, cf. [5]. FEX also gives covariance matrices for the extracted points and lines. The projection matrices were fixed though one can include statistical information here, too. For one scene we estimated 16 3D-lines corresponding to object edges, on average we used 4 point- and 4 line observation for each 3D line. The endpoints of the 2D line segments defined the endpoints of the 3D line segments. The length of the line segments were between 2 [m] and 12 [m], the average standard deviation at the line endpoints orthogonal to the 3D line segments were between 0.02 [m] and 0.16 [m], cf. fig. 3.



**Fig. 2.** First row: this is the cumulative histogram of the estimated  $\hat{\sigma}_0^2$  for the estimated point, line and plane resp. Second row: scatter diagram of the estimated point  $\hat{\mathbf{Y}}$  projected on the  $xy$ -,  $xz$ -,  $yz$ -planes. The inner confidence ellipse is averaged over all 1000 estimated ellipses, the outer one is the empirical confidence ellipse based on all estimated points and is approx. larger by a factor 1.2

## 7 Conclusions

We proposed a generic estimation method for estimating points, lines and planes from identical, incident, parallel or equal points, lines and planes. It can be used



**Fig. 3.** From four aerial image patches (left) we manually matched 2D features of 16 roof edges (middle). The average cross error  $\sigma_c$  at the endpoints of estimated 3D line segment doesn't exceed 0.17 [m], the length  $l$  of the line segments varied between 2 and 14 [m], see text for details.

in a wide variety of applications and does not need any approximate values for the unknowns. The implementation has been tested on a large artificial dataset to validate its correctness. Future work includes among other topics: (i) The estimation of polyhedral structures like polygons on a plane or cuboid elements. (ii) Additional constraints like metric constraints, e.g. fixed distance between entities. (iii) The estimation method could be embedded in a robust estimation scheme like RANSAC.

## References

1. R. Duda and P. Hart. Pattern classification and scene analysis, 1973. 1
2. O. Faugeras and T. Papadopoulo. Grassmann-cayley algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors. In *Trans. of the ROYAL SOCIETY A*, 365, pages 1123–1152, 1998. 1, 5
3. W. Förstner. On Estimating 2D Points and Lines from 2D Points and Lines. In *Festschrift anlässlich des 60. Geburtstages von Prof. Dr.-Ing. Bernhard Wrobel*, pages 69 – 87. Technische Universität Darmstadt, 2001. 5
4. W. Förstner, A. Brunn, and S. Heuel. Statistically testing uncertain geometric relations. In G. Sommer, N. Krüger, and Ch. Perwass, editors, *Mustererkennung 2000*, pages 17–26. DAGM, Springer, September 2000. 2.1, 2.2, 3, 1, 1
5. C. Fuchs. *Extraktion polymorpher Bildstrukturen und ihre topologische und geometrische Gruppierung*. DGK, Bayer. Akademie der Wissenschaften, Reihe C, Heft 502, 1998. 6
6. R.I. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000. 1, 2.2
7. K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, 1996. 1
8. E. M. Mikhail and F. Ackermann. *Observations and Least Squares*. University Press of America, 1976. 4
9. B. Steines and S. Abraham. Metrischer Trifokaltensor für die Auswertung von Bildfolgen. In W. Förstner, J.M. Buhmann, A. Faber, and P. Faber, editors, *Mustererkennung '99*, LNCS, 1999. 4