Matching, Reconstructing and Grouping 3D Lines From Multiple Views Using Uncertain Projective Geometry

Stephan Heuel, Wolfgang Förstner
heuel@ipb.uni-bonn.de, wf@ipb.uni-bonn.de
Institute for Photogrammetry, University of Bonn, Germany

Abstract

We present a geometric method for (i) matching 2D line segments from multiple oriented images, (ii) optimally reconstructing 3D line segments and (iii) grouping 3D line segments to corners.
The proposed algorithm uses two developments in combining projective geometry and statistics, which are described in this article: (i) the geometric entities points, lines and planes in 2D and 3D and their uncertainty are represented in homogeneous coordinates and new entities may be constructed including their propagated uncertainty. The construction can be performed directly or as an estimation. (ii) relations such as incidence, equality, parallelity and orthogonality between points, lines and planes can be tested statistically based on a given significance level.
Using these tools, the resulting algorithm is straightforward and gives reasonable results. It is only based on geometric information and does not use any image intensities, though it can be extended to use other information. The matching of 3D lines does not need any thresholds other than a significance value for the hypotheses tests.

1. Introduction

Reconstructing polyhedral objects from multiple views is an on-going task in computer vision with numerous applications, e.g. in building extraction, cf. [15]. The computation of straight 3D line segments has been shown to be a successful approach to the reconstruction of polyhedral objects, for example in [17] or [1]. Line matching is the first step to obtain 3D line segments by establishing correspondences of features between multiple images of a scene. In general it is a difficult problem because of cluttered and noisy image features and weaker geometric constraints compared to the task of point matching. Therefore in addition to the geometric constraints between views, additional informations such as length or orientation of the 2D lines segment and/or intensity based cues are used to compute similarity scores between matching hypotheses.

On the other hand, algebraic projective geometry has been extensively promoted in Computer Vision in the last decade (cf. [4],[9]), simplifying the representation of geometric entities. Using the consistent representation, it is possible to do geometric reasoning: (i) constructing new entities by intersecting or joining given entities and (ii) testing spatial relationships between geometric entities. However, to our knowledge, statistical spatial reasoning based on algebraic projective geometry covering construction, estimation and testing has not been presented up to now.

In this paper we first present a calculus of projective geometric entities containing (a) representations of projective entities with its uncertainties, (b) rules of construcing entities both non-redundant and redundant and (c) hypothesis tests of relations between projective entities. We then introduce an algorithm, which exploits only the geometric constraints in multiple views to solve the following tasks: (i) matching 2D line segments across multiple views, (ii) optimally reconstructing 3D lines from matched 2D lines, (iii) grouping 3D line segments to higher aggregates, in our case 3D corners consisting of a corner point and two 3D line segments.

The work closest to ours is the monograph by Kanatani, cf. [13] who presents techniques for statistical geometric reasoning using homogeneous representations, but does not make full use of the elegant projective formulations, leading to complex expressions. Also Clarke [2] and Criminisi [3] use covariance matrices of homogeneous entities and analyze the neglible effects of second order terms for mean and variance. The implicit representation with singular inverse covariance matrices given by Seitz [18] is only useful for intersection of points, lines and planes, not for the join.

The paper is organized as follows: section 2 introduces the homogeneous representation of points, lines and planes and their uncertainty. Section 3 shows how to construct new entities taking the uncertainty into account. The construction can be done either directly or by an iterative estimation. Both sections are based on our previous work [7], [11].

Section 4 shows how to test relations such as incidence or equality between geometric entities, section 5 describes the matching and reconstruction algorithm which we developed using the tools from the previous sections. Next, we demonstrate how to use these tools for grouping 3D line segments to corners. We give first results, summarize our approach and list some future work.

2. Uncertain Geometric Entities

We want to extend projective geometry in a manner which allows to reason with unertain geometric entities. Based on homogeneous vectors and matrices (sec. 2.1) we show how to represent uncertain homogeneous entities (sec.2.2). This will allow simple construction of uncertain projective entities (sec. 3.1) and direct estimation in case of redundant measurements (sec. 3.2).

2.1. Homogeneous Entities

We represent points, lines and planes in 2D and 3D with homogeneous vectors, cf. [9]. As listed in table 1, 3D points are denoted with $\mathbf{X}$, $\mathbf{Y}$, 3D lines with $\mathbf{L}$, $\mathbf{M}$ and planes with $\mathbf{A}$, $\mathbf{B}$. Points and lines in 2D are written in homogeneous vectors $\mathbf{x}^\mathsf{T} = (\boldsymbol{x}_0^\mathsf{T}, x_h) = (u, v;\ w)$ resp. $\mathbf{l}^\mathsf{T} = (\boldsymbol{l}_h^\mathsf{T}, l_0) = (a, b;\ c)$. Euclidean vectors will be denoted with italic bold letters like $\boldsymbol{x}$ for a Euclidean 2D point.

3D	vector/matrix
point	$\mathbf{X}^\mathsf{T} = (\boldsymbol{X}_0^\mathsf{T}, X_h) = (U, V, W;\ T)$
	$\underbrace{\mathsf{\Pi}^\mathsf{T}(\mathbf{X})}_{4\times 6} = \begin{pmatrix} X_h \boldsymbol{I} & \mathsf{S}(\boldsymbol{X}_0) \\ -\boldsymbol{X}_0^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{pmatrix}$
line	$\mathbf{L}^\mathsf{T} = (\boldsymbol{L}_h^\mathsf{T}, L_0^\mathsf{T}) = (L_1, L_2, L_3;\ L_4, L_5, L_6)$
	$\underbrace{\mathsf{\Gamma}(\mathbf{L})}_{4\times 4} = \begin{pmatrix} -\mathsf{S}(\boldsymbol{L}_h) & -\boldsymbol{L}_0 \\ \boldsymbol{L}_0^\mathsf{T} & 0 \end{pmatrix}$; $\overline{\mathsf{\Gamma}}(\mathbf{L}) = \mathsf{\Gamma}(\overline{\mathbf{L}})$
plane	$\mathbf{A}^\mathsf{T} = (\boldsymbol{A}_h^\mathsf{T}, A_0) = (A, B, C;\ D)$
	$\underbrace{\overline{\mathsf{\Pi}}^\mathsf{T}(\mathbf{A})}_{4\times 6} = \begin{pmatrix} -\mathsf{S}(\boldsymbol{A}_h) & A_0 \boldsymbol{I} \\ -\boldsymbol{A}_h^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{pmatrix}$

Table 1: *3D Points, lines, and planes represented as homogeneous vectors* $\mathbf{X}$, $\mathbf{L}$, $\mathbf{A}$ *(top) and matrices* $\mathsf{\Pi}^\mathsf{T}(\mathbf{X}), \mathsf{\Gamma}(\mathbf{L}), \overline{\mathsf{\Pi}}^\mathsf{T}(\mathbf{A})$ *(bottom). The skew-symmetric matrix* $\mathsf{S}(\boldsymbol{X}_0) = [\boldsymbol{X}_0]_\times$ *is the Jacobian of the vector product. The Plücker coordinates* L_i *are chosen such that for two Euclidean points* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *the line is given as* $\mathbf{L} = (\boldsymbol{Y} - \boldsymbol{X};\ \boldsymbol{X} \times \boldsymbol{Y})^\mathsf{T}$. *The matrix* $\mathsf{\Gamma}(\mathbf{L})$ *is the so-called Plückermatrix of a line* $\mathbf{L}$*; for points and planes we have the matrices* $\mathsf{\Pi}^\mathsf{T}(\mathbf{X})$ *and* $\overline{\mathsf{\Pi}}^\mathsf{T}(\mathbf{A})$

.

Each homogeneous vector contains a Euclidean part, indexed with a zero, and a homogeneous part, indexed with h. The indexing 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 the ratio of $d_{A,0} = |A_0|/|\boldsymbol{A}_h|$. Note that a line $\mathbf{L}^\mathsf{T} = (\boldsymbol{L}_h^\mathsf{T}, \boldsymbol{L}_0^\mathsf{T})$ is a 6-vector which has to fulfill the Plücker condition $\overline{\mathbf{L}}^\mathsf{T}\mathbf{L} = 0$ with the dual line $\overline{\mathbf{L}}^\mathsf{T} = (\boldsymbol{L}_0^\mathsf{T}, \boldsymbol{L}_h^\mathsf{T})$. The dual of a point is a plane and vice versa. In addition to the homogeneous vectors we also use homogeneous matrices $\mathsf{\Pi}(\mathbf{X}), \mathsf{\Gamma}(\mathbf{L})$ and $\overline{\mathsf{\Pi}}(\mathbf{A})$. Note that the matrices $\mathsf{\Pi}(\mathbf{X})$ and $\overline{\mathsf{\Pi}}(\mathbf{A})$ play the same role for points and planes as the Plücker-matrix $\mathsf{\Gamma}(\mathbf{L})$ for lines.

2.2. Uncertain Homogeneous Entities

We now want to extend the representation to be able to deal with uncertain geometric entities. In order to clarify our idea we first describe spatial reasoning with homogeneous vectors or matrices:

We argue that using homogeneous coordinates is just using the *idea* of projective geometry to be able to take advantage of the simple, mostly multilinear relations for spatial reasoning: we practically only use *single representatives of the equivalence classes of homogneous vectors* for spatial reasoning, derived from the Euclidean representation of the geometric entity. This allows to represent uncertain geometric entities as uncertain vectors in some $\mathbb{R}^\mathrm{k}$. Going back to a Euclidean representation requires at least some nomalization to get rid of the unknown scale of the homogeneous vector, cf. fig. 1.

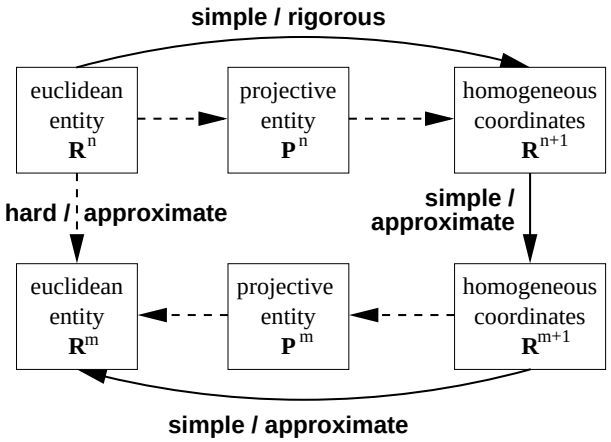

Figure 1: *Different ways of spatial reasoning using Euclidean or homogeneous coordinates. The reasoning using homogeneous coordinates can performed in projective space, e. g.leading from a set of entities in* $\mathbb{P}^\mathrm{n}$ *to an entity in* $\mathbb{P}^\mathrm{m}$. *The actual reasoning (1) starts from coordinate vectors, say in* $\mathbb{R}^\mathrm{n}$*, (2) transfers them to single homogeneous vectors in* $\mathbb{R}^{\mathrm{n+1}}$*, (3) yields a single homogeneous vector in* $\mathbb{R}^{\mathrm{m+1}}$ *and (4) goes back to a Euclidean coordinate vector in* $\mathbb{R}^\mathrm{m}$ *by normalization. The calculations can be either hard or simple; handling uncertainty can be either rigorous or approximate, see text for details.*

Formally we represent uncertain vectors by the pair $(\mathbf{x}, \mathbf{\Sigma}_{xx})$, where $\mathbf{\Sigma}_{xx}$ denotes the covariance matrix of the vector $\mathbf{x}$. This is a valid approximation as long the directional uncertainty of the homogeneous vector is small. Thus

the uncertainty of the direction of the spherically normalized vector $\mathbf{x}^s = \mathbf{x}/|\mathbf{x}|$ should be small.

This leads to the following valid *equivalence relations*: two uncertain homogeneous vectors $(\mathbf{x}, \mathbf{\Sigma}_{xx})$ and $(\mathbf{y}, \mathbf{\Sigma}_{yy})$ are equivalent if their spherically normalized versions are identical:

$$(\mathbf{x}, \mathbf{\Sigma}_{xx}) \equiv (\mathbf{y}, \mathbf{\Sigma}_{yy}) \;:\Leftrightarrow\; \mathbf{x}^s = \mathbf{y}^s \;\text{ and }\; \mathbf{\Sigma}_{x^s x^s} = \mathbf{\Sigma}_{y^s y^s}$$

The covariance matrices $\mathbf{\Sigma}_{x^s x^s}$ and $\mathbf{\Sigma}_{y^s y^s}$ can be derived by error propagation. Because of the normalization these covariance matrices are singular. On the other hand, the covariance matrix of an non-normalized uncertain homogeneous vector may in general have full rank, cf. fig. 2. As an example the 2D line $(\mathbf{l}, \mathbf{\Sigma}_{ll})$ joining two points $(\mathbf{x}, \mathbf{\Sigma}_{xx})$, $(\mathbf{y}, \mathbf{\Sigma}_{yy})$ is given by, cf. eq. (1)

$$(\mathbf{l}, \mathbf{\Sigma}_{ll}) = \left(\mathbf{x} \times \mathbf{y},\; \mathsf{S}(\mathbf{y})\mathbf{\Sigma}_{xx}\mathsf{S}^\mathsf{T}(\mathbf{y}) + \mathsf{S}(\mathbf{x})\mathbf{\Sigma}_{yy}\mathsf{S}^\mathsf{T}(\mathbf{x})\right)$$

with the skew-symmetric matrix $\mathsf{S}(\mathbf{x}) = [\mathbf{x}]_\times$, cf.[13], eq. (4.18). The covariance matrix $\mathbf{\Sigma}_{ll}$ is regular unless the two points are identical.

Now we can apply spatial reasoning not to homogeneous vectors $\mathbf{x}$, but to vector-covariance matrix pairs $(\mathbf{x}, \mathbf{\Sigma}_{xx})$. According to figure 1, spatial reasoning with homogeneous vectors requires two types of operations: (1) operations on homogeneous vectors such as join, intersection or image projection, and (2) normalization to euclidean entities. Due to the non-linearity of these operations, error propagation with covariance matrices is only an approximation compared to rigorous propagation with probability density functions (pdf). As spatial reasoning with homogeneous coordinates is based on bilinear forms (cf. table 2) and as normalization of homogeneous coordinates is a division, the analysis of the degree of approximation using homogeneous entities is much simpler than using only Euclidean entities, cf. [3] and [2].

Remarks. • We use the classical representation of uncertainty with Gaussian distributions as assumption for representing original measurement errors. This assumption in general is reasonable due to the central limit theorem. In certain cases one needs to analyze the degree of approximation based on the pdf, e.g. when using small distances between two Gaussian-distributed points.

• Rigorous uncertainy propagation of homogeneous entities using pdfs would require to handle generalized functions, because of the redundant representation of the homogeneous vectors. E. g. we have the classical representative $\mathbf{x}^\mathsf{T} = (\lambda \boldsymbol{x}^\mathsf{T}, \lambda)$,with $\lambda = 1$. We now interprete $\lambda = 1$ as a stochastical variable with $E(\lambda) = 1$ and $V(\lambda) = 0$, leading to the delta function $\delta(\lambda - 1)$ as the pdf of λ. This is a generalized function. The distribution of λ is the step function being a proper function. Finally, the distribution of the homogeneous vector $(\mathbf{x}, 1)$ is given by $P_{\mathbf{x}}(\mathbf{x}) = P_x(\boldsymbol{x}) \cdot s(\lambda - 1)$. In case we can assume the Euclidean vector $\boldsymbol{x}$ to be Gaussian distributed with covariance matrix $\mathbf{\Sigma}_{xx}$ the homogeneous vector $\mathbf{x}^\mathsf{T} = (\boldsymbol{x}^\mathsf{T}, 1)$ has covariance matrix $\mathbf{\Sigma}_{\mathbf{xx}} = \begin{pmatrix} \mathbf{\Sigma}_{xx} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{pmatrix}$ which is singular allowing simple error propagation.

• We use the Laplacian approximation by cutting off the Taylor series after the linear term. This causes systematic errors (bias) in mean and variance. However one can show that for bilinear constructions and for normalizations this bias is small, namely much smaller than the standard deviation of the mean or the standard deviation of the variance, if the relative accuracy is better than 5 % which will be nearly always the case, see also [13],[3],[2]. For bilinear forms only in case the standard deviation of both entities is larger than their mean, one has to expect significant errors, cf. [8].

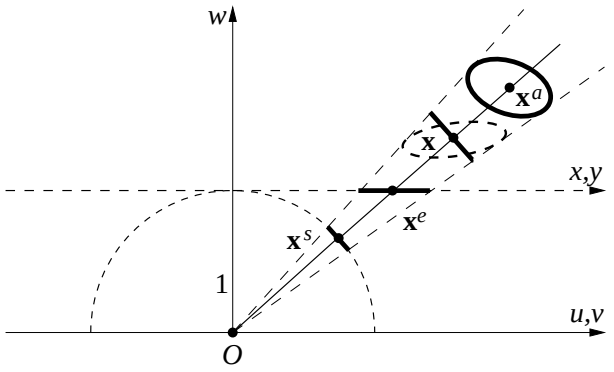

Figure 2: *Equivalence class of homogeneous covariance matrices for a projective point. Observe that the covariance matrices may be regular as for the arbitrary point $\mathbf{x}^a$ or singular, especially for the vector $\mathbf{x} = (u, v, w)^\mathsf{T}$ and $\mathbf{x}^s$ and the vector $\mathbf{x}^e = (x, y, 1)^\mathsf{T}$ with Euclidean normalization.*

3. Construction of Uncertain Entities

3.1. Direct Construction

First we show how to construct points, lines and planes directly using intersection and join of given entities, the four combinations with two given entities are listed in table 2 or cf. [7]. Using homogeneous vectors, these constructions are a matter of a matrix-vector multiplication. Furthermore we can back-project 2D points and lines from the image to object space: given an image line $\mathbf{l}'$ and a projection matrix P, one can construct the projecting plane $\mathbf{A}'(\mathbf{l}') = \mathsf{P}^\mathsf{T}\mathbf{l}'$, cf. fig. 3. Similarly, back-projecting an image point $\mathbf{x}'$ yields the projecting line $\mathbf{L}'(\mathbf{x}') = \mathsf{Q}^\mathsf{T}\mathbf{x}'$, where the 3×6 line projection matrix Q can be computed directly from P, cf. [9].

All constructions listed in table 2 are bilinear functions

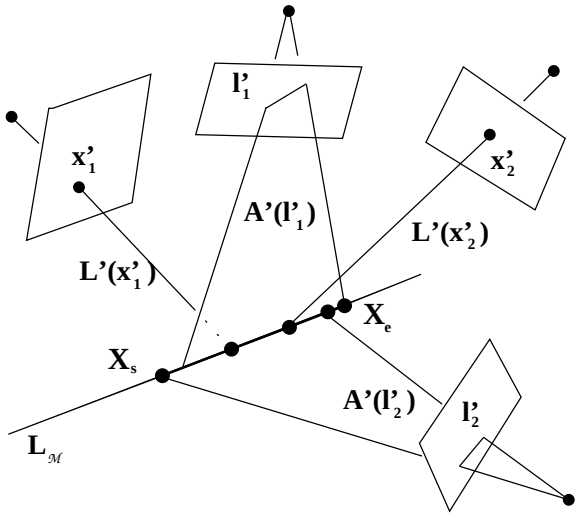

Figure 3: *shows the ideal geometric situation, where a 3D line* $\mathbf{L}_{\mathcal{M}}$ *can be constructed by a set of matched image features* $\mathcal{M} = \{\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{l}'_1, \mathbf{l}'_2\}$. *For each image point, we can define a projecting line* $\mathbf{L}'(\mathbf{l}'_{1/2})$ *and for each image line segment a projecting plane* $\mathbf{A}'(\mathbf{x}'_{1/2})$. *A 3D line segment* $\mathbf{S}$ *is defined by the pair of points* $(\mathbf{X}_s, \mathbf{X}_e)$ *on the line* $\mathbf{L}_{\mathcal{M}}$ *with the largest distance to each other.*

on its given entities, i.e. in general $\underline{\mathbf{z}} = \mathsf{U}(\underline{\mathbf{y}})\underline{\mathbf{x}} = \mathsf{V}(\underline{\mathbf{x}})\underline{\mathbf{y}}$, where $\mathsf{U}(\underline{\mathbf{y}})$ and $\mathsf{V}(\underline{\mathbf{x}})$ are matrices with entries linearly depending on $\underline{\mathbf{y}}$ and $\underline{\mathbf{x}}$. Assuming that $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$ are uncorrelated, we can use the standard error propagation to obtain the uncertainty of $\underline{\mathbf{z}}$:

$$\mathbf{\Sigma}_{zz} = \mathsf{U}(\mathbf{y})\mathbf{\Sigma}_{xx}\mathsf{U}^{\mathsf{T}}(\mathbf{y}) + \mathsf{V}(\mathbf{x})\mathbf{\Sigma}_{yy}\mathsf{V}^{\mathsf{T}}(\mathbf{x}) \qquad (1)$$

Therefore by having the covariances of the given entities, we can easily do error propagation to obtain the uncertainty of the newly constructed entity.

3.2. Estimation

In the previous section we presented direct constructions of uncertain points, lines and planes, but in general we might have a large number of observed entities being either incident ($\in$) or identical ($\equiv$) to the unknown entity. For example, in fig. 3, there are four cameras with two projecting lines and two projecting planes incident to the unknown space line.

Gauß-Helmert model. To estimate the unknown entity, we use an iterative, linear estimation model, the so-called Gauß-Helmert model (cf. [10],[16]) which is summarized as follows (cf. [6]): the constraints between the true unknown

entities	construction	expression
$\mathbf{X}, \mathbf{Y}$	$\mathbf{L} = \mathbf{X} \wedge \mathbf{Y}$	$\mathbf{L} = \overline{\mathsf{\Pi}}(\mathbf{X})\mathbf{Y} = -\overline{\mathsf{\Pi}}(\mathbf{Y})\mathbf{X}$
$\mathbf{A}, \mathbf{B}$	$\mathbf{L} = \mathbf{A} \cap \mathbf{B}$	$\mathbf{L} = \overline{\mathsf{\Pi}}(\mathbf{A})\mathbf{B} = -\overline{\mathsf{\Pi}}(\mathbf{B})\mathbf{A}$
$\mathbf{X}, \mathbf{L}$	$\mathbf{A} = \mathbf{X} \wedge \mathbf{L}$	$\mathbf{A} = \overline{\mathsf{\Pi}}^{\mathsf{T}}(\mathbf{X})\mathbf{L} = -\mathsf{\Gamma}(\mathbf{L})\mathbf{X}$
$\mathbf{A}$,$\mathbf{L}$	$\mathbf{X} = \mathbf{A} \cap \mathbf{L}$	$\mathbf{X} = \mathsf{\Pi}^{\mathsf{T}}(\mathbf{A})\mathbf{L} = -\overline{\mathsf{\Gamma}}(\mathbf{L})\mathbf{A}$
$\mathbf{l}', \mathsf{P}$	$\mathbf{A}'(\mathbf{l}', \mathsf{P})$	$\mathbf{A}' = \mathsf{P}^{\mathsf{T}}\mathbf{l}' = (\mathbf{l}'^{\mathsf{T}} \otimes I_4)\mathrm{vec}(\mathsf{P}^{\mathsf{T}})$
$\mathbf{x}', \mathsf{Q}$	$\mathbf{L}'(\mathbf{x}', \mathsf{Q})$	$\mathbf{L}' = \mathsf{Q}^{\mathsf{T}}\mathbf{x}' = (\mathbf{x}'^{\mathsf{T}} \otimes I_6)\mathrm{vec}(\mathsf{Q}^{\mathsf{T}})$

Table 2: *Direct construction of new points* $\mathbf{X}$, $\mathbf{Y}$, *lines* $\mathbf{L}, \mathbf{M}$ *or planes* $\mathbf{A}, \mathbf{B}$ *using join* $\wedge$ *and intersection* $\cap$. *The matrices* $\mathsf{\Pi}$, $\overline{\mathsf{\Pi}}$, $\mathsf{\Gamma}$ *and* $\overline{\mathsf{\Gamma}}$ *are given in table 1.* P *and* Q *are the projection matrices for points resp. lines. With image points* $\mathbf{x}'$ *and image lines* $\mathbf{l}'$ *we can construct the projecting line* $\mathbf{L}'$ *and the projecting plane* $\mathbf{A}'$. *The symbol* $\otimes$ *denotes the Kronecker product, the operator* vec *appends the columns of a matrix to a vector. All forms are linear in the coordinates of the given entities allowing rigorous error propagation.*

entity $\tilde{\boldsymbol{\beta}}$ and a list of ideal observed entities $\tilde{\boldsymbol{\gamma}}_i$ expressed as an implicit form $\boldsymbol{g}_i(\tilde{\boldsymbol{\beta}};\ \tilde{\boldsymbol{\gamma}}_i) = \mathbf{0}$, with constraints $\boldsymbol{h}(\tilde{\boldsymbol{\beta}}) = 0$ on the unknown. The actual observations $\underline{\boldsymbol{\gamma}}_i$ are random perturbations of the ideal observations $\tilde{\boldsymbol{\gamma}}_i$: $\underline{\boldsymbol{\gamma}}_i = \tilde{\boldsymbol{\gamma}}_i + \underline{\boldsymbol{e}}_i$, where $\underline{\boldsymbol{e}}$ is assumed to be normally distributed with mean $\mathbf{0}$ and covariance matrix $\underline{\mathbf{\Sigma}}_{ee} = \sigma^2\underline{\mathbf{\Sigma}}^i_{ee}$. Minimizing the weighted sum of the residuals of the constraints we obtain an estimation $(\hat{\boldsymbol{\gamma}}_i, \hat{\boldsymbol{\beta}})$. Furthermore we obtain a covariance matrix $\hat{\mathbf{\Sigma}}_{\hat{\beta}\hat{\beta}} = \hat{\sigma}^2\mathbf{\Sigma}^i_{\hat{\beta}\hat{\beta}}$ of the unknown entity with the estimated initial covariance matrix $\mathbf{\Sigma}^i_{\hat{\beta}\hat{\beta}}$ and the estimated variance factor $\hat{\sigma}^2$. which is an indication how well the estimate fits to the given data and its covariances.

Application. In our case we need an implicit algebraic expression of the form $\boldsymbol{g}_i(\boldsymbol{\beta};\ \boldsymbol{\gamma}_i) = \mathbf{0}$ for expressing incidence and equality of points, lines and planes, all algebraic expressions can be found in table 3. For example, if an unknown line $\mathbf{M}$ should be incident to observed planes $\mathbf{A}_i$, then the following constraint holds: $\boldsymbol{g}(\mathbf{M};\ \mathbf{A}_i) = \mathsf{\Pi}^{\mathsf{T}}(\mathbf{A}_i)\mathbf{M} = -\mathsf{\Gamma}(\mathbf{M})\mathbf{A}_i = \mathbf{0}$. Since all constraints in table 3 are bilinear in both the observations and unknowns, we already have the Jacobians necessary for the estimation procedure.

Using these equations, the estimation of points, lines and planes from incident and identical entities can be performed with the previously described Gauss-Helmert model. Since we use an iterative estimation procedure, we need approximate values, which can be obtained directly using eigenvector or SVD solutions. In [11], it is shown how to extend this estimation scheme with parallelity and orthogonality constraints for 2D and 3D objects yielding good results within a few iterations. For an alternative formulation of the fitting

observations→ ↓ unknown	points $\mathbf{X}_i$		lines $\mathbf{L}_i$		planes $\mathbf{A}_i$	
	relation	constraint	relation	constraint	relation	constraint
point $\mathbf{Y}$	$\mathbf{X}_i \equiv \mathbf{Y}$	$\overline{\mathsf{\Pi}}(\mathbf{X}_i)\,\mathbf{Y} = \mathbf{0}$	$\mathbf{L}_i \ni \mathbf{Y}$	$\overline{\mathsf{\Gamma}}(\mathbf{L}_i)\,\mathbf{Y} = \mathbf{0}$	$\mathbf{A}_i \ni \mathbf{Y}$	$\mathbf{A}_i^\mathsf{T}\mathbf{Y} = 0$
line $\mathbf{M}$	$\mathbf{X}_i \in \mathbf{M}$	$\overline{\mathsf{\Pi}}^\mathsf{T}(\mathbf{X}_i)\,\mathbf{M} = \mathbf{0}$	$\mathbf{L}_i \equiv \mathbf{M}$ $\mathbf{L}_i \cap \mathbf{M}_i \neq \emptyset$	$\overline{\mathsf{\Gamma}}(\mathbf{L}_i)\,\mathsf{\Gamma}(\mathbf{M}) = \mathsf{0}$ $\overline{\mathbf{L}_i}^\mathsf{T}\,\mathbf{M} = 0$	$\mathbf{A}_i \ni \mathbf{M}$	$\mathsf{\Pi}^\mathsf{T}(\mathbf{A}_i)\,\mathbf{M} = \mathbf{0}$
plane $\mathbf{B}$	$\mathbf{X}_i \in \mathbf{B}$	$\mathbf{X}_i^\mathsf{T}\,\mathbf{B} = 0$	$\mathbf{L}_i \in \mathbf{B}$	$-\overline{\mathsf{\Gamma}}(\mathbf{L}_i)\,\mathbf{B} = \mathbf{0}$	$\mathbf{A}_i \equiv \mathbf{B}$	$\overline{\mathsf{\Pi}}(\mathbf{A}_i)\mathbf{B} = \mathbf{0}$

Table 3: *Possible equality and incidence relations $\equiv, \in, \cap \neq \emptyset$ between lists of observed space points $\mathbf{X}_i$, lines $\mathbf{L}_i$, and planes $\mathbf{A}_i$ (top column) and unknown space points $\mathbf{Y}_i$, lines $\mathbf{M}_i$ and planes $\mathbf{B}_i$ (left row). The algebraic formulation of the constraints are bilinear with respect to the unknowns and observations. The matrices $\mathsf{\Pi}$, $\overline{\mathsf{\Pi}}$, $\mathsf{\Gamma}$ and $\overline{\mathsf{\Gamma}}$ are given in table 1.*

problem, see e.g. [19].

4. Testing Geometric Relations

Testing whether two points are identical or whether a line is incident to another line can be formulated as an hypothesis test; in fact, all relations between points, lines and planes given in table 4 can be used. In the following we assume all variables to be normally distributed. We use the following theorem from statistical testing theory (cf. [14]):

Test of $\boldsymbol{d} = \mathbf{0}$: Given an n-vector $\boldsymbol{d}$ with normal distribution $\underline{\boldsymbol{d}} \sim N(\mathbf{0}, \mathbf{\Sigma})$, rk$\mathbf{\Sigma} = r \leq n$, and known null-space $\mathcal{N}(\mathbf{\Sigma}) = \mathsf{H}$, being a $n \times (n-r)$-matrix, the the optimal test statistic for the hypothesis $H_o : \boldsymbol{x} = \mathbf{0}$ is given by

$$\underline{T} = (\underline{\boldsymbol{x}} - \boldsymbol{\mu})^\mathsf{T}\mathbf{\Sigma}^+(\underline{\boldsymbol{x}} - \boldsymbol{\mu}) \sim \chi^2_r \qquad (2)$$

where χ^2_r denotes the χ^2_r-distribution with r degrees of freedom. For the pseudo inverse $\mathbf{\Sigma}^+$ we use the null-space H, which can be determined algebraically for our cases, cf. [7]. Therefore we choose a significance number α and compare T with the critical value $\chi^2_{r,\alpha}$. If $T > \chi^2_{r,\alpha}$ then the hypothesis that the spatial relation holds can be rejected.

In table 3, the expressions for computing the homogeneous vector $\mathbf{d}$ are given: $\mathbf{d}$ is supposed to be zero if the incidence or equality relation for two entities is fulfilled. Note that for testing two elements it is advantageous to scale down the Euclidean part w.r.t. the homogeneous part to minimize linearization errors of the covariance matrices. Including parallelity and orthogonality relations we can formulate 13 relationships for 3D entities and therefore 13 different hypotheses tests, cf. table 4 and [7].

5. Matching and Reconstruction

We now describe the approach to match 2D line segments and reconstruct 3D lines based on the previously described tools for geometric reasoning and construction taking the uncertainties of the observations into account.

	Point $\mathbf{X}$	Line $\mathbf{L}$	Plane $\mathbf{A}$
Point $\mathbf{Y}$	$\in$	$\in$	$\in$
Line $\mathbf{A}$		$\cap \neq \emptyset$, $\equiv$, $\|$, $\perp$	$\in$, $\|$, $\perp$
Plane $\mathbf{B}$			$\equiv$, $\|$, $\perp$

Table 4: *13 relationships between points, lines and planes in 3D: incidence (denoted by $\in, \cap \neq \emptyset$), equality ($\equiv$), orthogonality ($\perp$) and parallelity relations ($\|$). Our system can perform hypotheses tests on these relations based on covariances, cf. [7]. The bilinear algebraic expressions for incidence and equality constraints used in this work are listed in table 3.*

Input data. For our task, the observations are points $\mathbf{x}$ and line segments $\mathbf{s}$, found by a point and line segment extractor, collected in N sets of features $\mathcal{F}^i$ for $i = 1, \ldots, N$ images. We use the polymorphic feature extractor, cf. [5], which gives - in addition to the geometric position - covariance matrices as uncertainty measures. From a 2D line segment $\mathbf{s}$ one can derive the (infinite) 2D line $\mathbf{l}(\mathbf{s})$ with $\mathbf{\Sigma}_{ll}$. We assume that the feature extraction yields realistic estimates for $\mathbf{\Sigma}_{xx}$ and $\mathbf{\Sigma}_{ll}$. If not, one has to calibrate the variance factors, for example with manual matches in a bundle adjustment.

Furthermore the projection matrices of the images are given, possibly with their covariances.

Epipolar Beam. The matching algorithm uses only geometric constraints, starting with a constraint from epipolar geometry, cf. [20]: given a line segment $\mathbf{s}$ with start- end endpoint $\mathbf{x}_s, \mathbf{x}_e$, the epipolar lines of $\mathbf{x}_s$ and $\mathbf{x}_e$ define a region ("epipolar beam") in the other images. Therefore we define a set $\mathcal{E}(\mathbf{s})$ for each line segment $\mathbf{s}$, which contains all line segments *and* points being in the epipolar beam. For the epipolar beam, the uncertainty of $\mathbf{x}_s, \mathbf{x}_e$ has a small, negligible influence on the result and safely can be ignored.

Matching Algorithm. The main matching algorithm is outlined as follows, cf. pseudo-code below. For a pair of 2D lines $\mathbf{l}_1, \mathbf{l}_2$ we compute the 3D line $\mathbf{L}_{12}$ by intersecting the projecting planes $\mathbf{A}'(\mathbf{l}_1), \mathbf{A}'(\mathbf{l}_2)$ and match all features in the other image with $\mathbf{L}_{12}$ by incidence tests with the back-projected image features (lines 4-10). We did the tests in 3D, but it is equivalent to project $\mathbf{L}_{12}$ into the images and test on collinearity in 2D.

Then we test the consistency of the matched features in each image separately, (lines 12-16): all matched 2D points and 2D line segments in one image should be collinear to each other. We pairwise test collinearity between the image features, and remove a feature if the collinearity test was not successful and the feature has a larger test-statistic (eq. (2)) for the incidence with $\mathbf{L}_{12}$ than its counterpart.

Now a 3D line $\hat{\mathbf{L}}_{\mathcal{M}}$ is estimated from all point and line segment matches and check the estimated variance factor $\hat{\sigma}$ (lines 19-20). If the feature extraction was calibrated correctly, this factor should be equal to 1. The variance factor is fisher distributed; we allow a factor of up to 2, which basically is equivalent to an hypotheses test with a sufficiently large significance level. We finally check again if all matches in $\mathcal{M}$ are still incident to $\hat{\mathbf{L}}_{\mathcal{M}}$ (line 22).

The last part of the algorithm is the computation of the endpoints $(\mathbf{X}_s, \mathbf{X}_e)$ of the 3D line segment $\hat{\mathbf{S}}$ which lie on the estimated 3D line (line 24). This can be done by computing the projecting lines for all points and endpoints of the matched features $\mathcal{M}$ and collect the intersections with the estimated line $\hat{\mathbf{L}}_{\mathcal{M}}$. The two most distant intersection points are the endpoints of the new 3D line segment $\hat{\mathbf{S}}(\mathcal{M}, \hat{\mathbf{L}}_{\mathcal{M}})$, also cf. fig. 3.

It is possible that multiple 3D line segments $\hat{\mathbf{S}}_j$ have been extracted for the same true line segment $\tilde{\mathbf{S}}$. Then all $\hat{\mathbf{S}}_j$ should be equivalent to each other, which again is tested statistically. Currently we delete all identical 3D line segments except of one, but we can also unify them to a possible longer line segment.

```
1  for s1 ∈ F^1 ∪ ··· ∪ F^N do
2      for s2 ∈ E(s1) do    l1 := l(s1), l2 := l(s2)
3          // direct construction using first two lines
4          (L12, Σ_{L12L12}) := A'(l1) ∩ A'(l2)
5          M := ∅    // init match-set M
6          // search projections of L12 in other image
7          for x, m ∈ E(l1) \ (F(l1) ∪ F(l2)) do
8              if L'(x) ∩ L12 ≠ ∅ then x ∈ M fi
9              if A'(m) ∋ L12 then m ∈ M fi
10         od
11         // pairwise collinearity check of matched features
12         for i := 1, ..., N; x, m1/2 ∈ M ∩ F^i do
13             if (x ∉ m1/2 or ¬(m1 ≡ m2))
14                 then M := M \ {x or m1 or m2}
15             fi
16         od
17         skip M if ∃i : F^i ∩ M = ∅
18         // estimate 3D line by incident matches M
19         (L̂_M, Σ̂_LL, σ̂) := Estimate_Incident(M)
20         skip M if σ̂ > T_σ    // check estimation
21         // final check of matches with estimated L̂_M
22         if (∀x, m ∈ M :
23             A'(m) ∈ L̂_M and L'(x) ∩ L̂_M ≠ ∅)
24             then compute 3D line segment Ŝ(M, L̂_M)
25         fi
26     od
27 od
```

Algorithm 1 *to compute 3D line segments $\hat{\mathbf{S}}_j$ from feature sets $\mathcal{F}^i$,$i = 1, \ldots, N$ for N images. A feature set consists of image line segments $\mathbf{s}_{1,2}, \mathbf{m}_{1,2}$ and points $\mathbf{x}$. The set $\mathcal{E}(\mathbf{s})$ is the set for a line segment $\mathbf{s}$, which contains all line segments and points being in the epiploar beam of $\mathbf{s}$. The threshold T_σ is chosen according to an hypothesis test, see text for details. All tests and computation are done using the uncertainty of the entities.*

Remarks. The sketched algorithm is a first proposal for the extensive use of uncertain projective geometry for matching algorithm and has some interesting properties.

- Note that there are no parameters to be set other than significance levels for the hypotheses tests, which was fixed to 95%.
- The step of optimal reconstruction of a 3D line segments with respect to the observations is an essential part of the matching algorithm.
- Image points are integrated into the matching and contribute to the final 3D line segment.
- We currently do not use the trifocal tensor for finding matches in other images, but it is possible to integrate it into the search.
- Theoretically three images are enough to use our approach, but one can expect that having four or more images greatly improves the performance of the algorithm.
- The algorithm was tested using known camera matrices; but since all relations that are tested are invariant under projective transformation, one can use any set of projectively equivalent camera matrices.

6. Grouping

Being able to do statistically uncertain geometric reasoning not only enables us to perform matching of 2D line segments: we can also use the hypotheses tests of sec. 4 to group the computed 3D line segments to more complex aggregates. To do reasonable grouping we assume our object model to be polyhedral, consisting of straight lines, corner-points and planar surfaces. Here we introduce the first step of detecting 3D corners, i.e. corner-points associated with

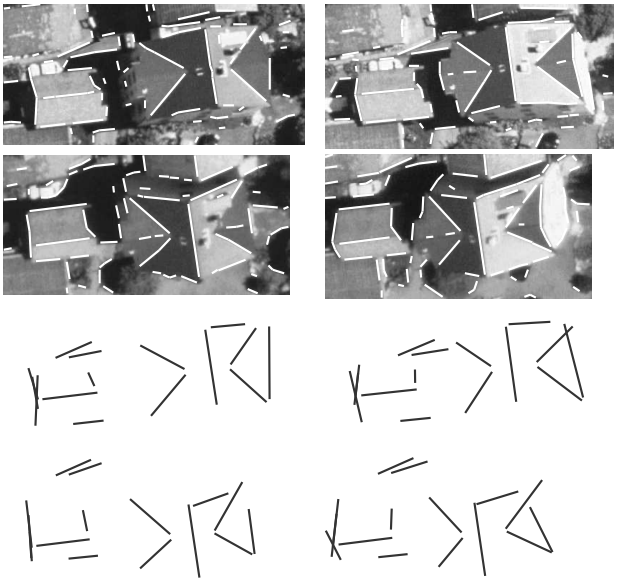

Figure 4: *Upper two rows: extracted line segments superimposed on four images; lower two rows: back-projected 3D line segments computed with the proposed algorithm using only line segments and no points.*

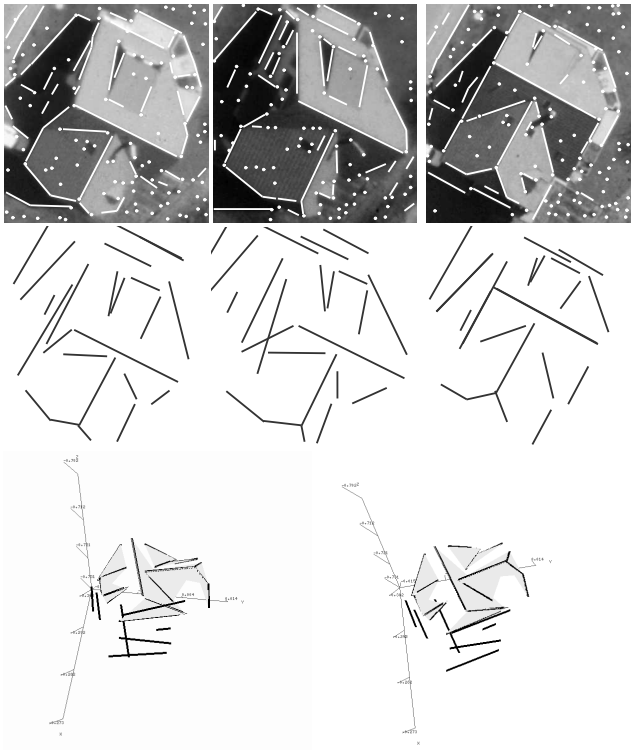

Figure 5: *first row: extracted line segments and points superimposed on three out of four images; second row: back-projected 3D line segments computed with the proposed algorithm using line segments and points; third row: two new views of the new 3D line segments, fourth rows two views with grouped corners.*

two 3D line segments. This means we have to find line segment pairs $\mathbf{S}_1, \mathbf{S}_2$ in object space $\mathcal{O}$ such that the lines $\mathbf{L}(\mathbf{S}_1), \mathbf{L}(\mathbf{S}_2)$ are incident and non-parallel, which can be tested statistically, as seen above. Furthermore we do not want to group line segments, that are too far away. If we know the units in object space, setting a threshold T_d is easy. The resulting grouping algorithm for finding a set $\mathcal{C}$ of 3D corners is given as:

```
1 for S1, S2 ∈ O do    L1 := L(S1), L2 := L(S2)
2   if L1 ∩ L2 ≠ ∅ and ¬(L1 || L2)
3     then Xc := Estimate_Incident(L1, L2)
4          skip if distance(Xc, S1, S2) > Td
5          C := C ∪ {(Xc, S1, S2)}
6   fi
7 od
```

In a more general grouping approach for 3D entities (cf. [12]), the same geometrical reasoning was used, but the distance criterion above was replaced with topological analysis of the features in image space. The resulting algorithm continued the outlined grouping step to polyhedral surfaces using coplanarity tests (cf. sec. 4) for corners.

7. Results

We tested the algorithm described in sec. 5 on aerial imagery, though there is no reason that the same algorithm works on close range scenes and we will test this in the near future. We included two example results, see figure 4 and 5; the first one only uses 2D line segments, the second one uses both line segments and points extracted by the local implementation of the feature extraction, cf.[5]. Example 1 contains 583 image lines and we obtained 15 3D line segments. Example 2 contains 181 2D segments and 443 points resulting in 23 3D line segments. On average, each estimated 3D line had about two point observations and four line observations.

The algorithm was implemented in the scripting language Perl; on a Pentium III it takes roughly 1 minute or less to compute 10 iterations in the outer loop of the algorithm in sec. 5, that means computing at best up to 10 3D line segments per minute. It appears to be sufficient to use only 50% or less of all existing image line segments as a starting segment $\mathbf{s}_1$, so that for the example in figure 4 we need between 15 minutes and half an hour to compute the 3D line segments. One may pre-sort the set of image line segments by their length so that more significant lines are used at the beginning.

8. Summary and Conclusions

We have presented a method for matching 2D line segments in multiple views and optimally reconstruct 3D line segments from these matches using an ML-type estimation scheme. The method only used geometrical constraints with error propagation and statistical tests, driven by the estimated precision of the image features. No data-dependent thresholds were used other than significance levels for hypotheses tests. Though it is desirable to use as many cues as possible, it has been shown that the geometric cue is quite strong and can be used if four images are given.

Furthermore it is possible to use the tools for estimation of geometric entities and testing its geometrical relations for grouping 3D elements. As an example we have grouped the 3D line segments to 3D corners.

Future work includes among other topics: (i) extending the estimation described in sec. 3.2 to a robust estimation, such that a feature within a match can be identified as an outlier. (ii) including other cues for matching and grouping of 3D lines which would stabilize the solution. Additionally, other cues would also improve the speed of the program, since a good match hypotheses is found earlier than with only geometric cues. As an example, one can use the 2D neighborhood relationship of points, lines and image regions to infer a 3D neighborhood relationship of 3D line segments, cf [12]. Other cues such as image intensities may also be used, cf. [17].

References

[1] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3D reconstruction of buildings from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, IAPRS Vol.32, Part 3-2W5*, pages 69–80, September 1999.

[2] J.C. Clarke. Modelling uncertainty: A primer. Technical Report 2161/98, University of Oxford, Dept. Engineering Science, 1998.

[3] A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. Springer-Verlag London Ltd., August 2001.

[4] 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.

[5] W. Förstner. A Framework for Low Level Feature Extraction. In J. O. Eklundh, editor, *Computer Vision - ECCV 94, Vol. II*, volume 802 of *LNCS*, pages 383–394. Springer, 1994.

[6] W. Förstner. Choosing Constraints in Image Triplets. In David Vernon, editor, *Computer Vision - ECCV 2000*, volume 1843, II of *Lecture Notes in computer science*, pages 669–684. Springer, 2000.

[7] 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.

[8] J. Haddon and D. A. Forsyth. Noise in bilinear problems. In *Proceedings of ICCV*, volume II, pages 622–627, Vancouver, July 2001. IEEE Computer Society.

[9] R.I. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[10] F. R. Helmert. *Die Ausgleichungsrechnung nach der Methode der Kleinsten Quadrate*. Teubner, Leipzig, 1872.

[11] S. Heuel. Points, lines and planes and their optimal estimation. In *Pattern Recognition, 23rd DAGM Symposium*, number 2191 in LNCS, pages 92–99. Springer, September 2001.

[12] S. Heuel, F. Lang, and W. Förstner. Topological and geometrical reasoning in 3d grouping for reconstructing polyhedral surfaces. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXIII, B3, pages 397–404, Amsterdam, 2000. ISPRS.

[13] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, 1996.

[14] K.-R. Koch. *Parameter estimation and hypothesis testing in linear models*. Springer, 1988.

[15] H. Mayer. Automatic object extraction from aerial imagery - a survey focusing on buildings. *Computer Vision and Image Understanding*, 74(2):138–149, 1999.

[16] E. M. Mikhail and F. Ackermann. *Observations and Least Squares*. University Press of America, 1976.

[17] Th. Moons, D. Frere, J. Vandekerckhove, and L. Van Gool. Automatic modelling and 3d reconstruction of urban house roofs from high resolution aerial imagery. *Computer Vision – ECCV'98, Freiburg*, pages 410–425, 1998.

[18] S. M. Seitz and P. Anandan. Implicit representation and scene reconstruction from probability density functions. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 28–34. IEEE Computer Society, 1999.

[19] B. Triggs. A new approach to geometric fitting. Submitted to 1998 International Conference on Computer Vision, January 1998.

[20] Z. Zhang. Token tracking in a cluttered scene. *Image and Vision Computing*, 12(2):110–120, 1194.