# Fast Statistically Geometric Reasoning About Uncertain Line Segments in 2D- and 3D-Space

Christian Beder

Institute of Photogrammetry, University of Bonn, Germany

**Abstract.** This work addresses the two major drawbacks of current statistical uncertain geometric reasoning approaches. In the first part a framework is presented, that allows to represent uncertain line segments in 2D- and 3D-space and perform statistical test with these practically very important types of entities. The second part addresses the issue of performance of geometric reasoning. A data structure is introduced, that allows the efficient processing of large amounts of statistical tests involving geometric entities. The running times of this approach are finally evaluated experimentally.

## 1 Introduction

In [5] the uncertain geometric entities point, line and plane in 2D- and 3D-space, represented using Grassmann-Cayley algebra, were used to perform statistical tests such as incidence, equality, parallelism or orthogonality between a pair of two entities. This is a very useful tool in many computer vision and perceptual grouping tasks, as both often deal with measurements of geometric entities and rely on the relational properties of the measured entities between each other (cf. [11], [8], [9]).

However, there are two major drawbacks in this approach: first the Grassmann-Caley algebra does not allow to represent localized objects, such as line segments in 2D- and 3D-space, in a straightforward manner and second there are no considerations about performing a huge amount of relational tests in an efficient manner.

Both of these drawbacks are addressed in this work. The first issue is addressed by using compound entities, i.e. to construct new geometric entities from the existing base entity classes, on the one hand and moving from the projective framework of [5] and [7] to an oriented projective framework (cf. [13]) on the other hand. The second issue is addressed by proposing a data structure for storing the entities and gaining efficiency in testing geometric relations over a large amount of data. The proposed data structure will resolve the shortcomings of the classical multi-dimensional data structures R-Tree, R*-Tree and Quadtree (cf. [6], [12],[4], [10]), that are unable to store uncertain line segments for the efficient use in statistical testing tasks.

The speed gained by using the proposed index structure for geometric reasoning, instead of simply computing all relational properties pairwise in a sequential manner, will be evaluated experimentally.

## 2 Compound Geometric Entities and their Relations

### 2.1 Base Entities in Oriented Projective Space

The line segments will be constructed from uncertain geometric base entities in oriented projected space. For this base entities first consider the 2D-case: a point and a line may be represented by homogeneous 3-vectors $\mathbf{x}$ and $\mathbf{l}$. In oriented projective space the sign of the scalar product $\mathbf{l}^T\mathbf{x}$ can be used to indicate, if the point lies on the right hand side or on the left hand side of the line. This can be used to define the notion of direction for lines and orientation for points. Note, that points with negative orientation do not correspond to Euclidean points. If one represents the uncertainty of the entities with their covariance matrices $\mathbf{\Sigma_{ll}}$ and $\mathbf{\Sigma_{xx}}$ and chooses a threshold $T_\alpha$ according to the $\chi^2$-distribution as proposed in [5], the statistical incidence test can be extended in the following way:

- if
$$\frac{\mathbf{l}^T\mathbf{x}}{\mathbf{l}^T\mathbf{\Sigma_{xx}}\mathbf{l}+\mathbf{x}^T\mathbf{\Sigma_{ll}}\mathbf{x}} < \sqrt{T_\alpha} \tag{1}$$
holds, there is no reason to reject the hypothesis, that the point lies on the left hand side of the line. This will be denoted by $\mathbf{x} \in^- \mathbf{l}$.
- if
$$-\sqrt{T_\alpha} < \frac{\mathbf{l}^T\mathbf{x}}{\mathbf{l}^T\mathbf{\Sigma_{xx}}\mathbf{l}+\mathbf{x}^T\mathbf{\Sigma_{ll}}\mathbf{x}} \tag{2}$$
holds, there is no reason to reject the hypothesis, that the point lies on right hand side of the line. This will be denoted by $\mathbf{x} \in^+ \mathbf{l}$.

Notice, that the two cases are not mutually exclusive, but the combination of both conditions yields the classical incidence relation, that is proposed in [5]. This will be denoted by $\mathbf{x} \in \mathbf{l}$.

In 3D-space the situation for points and planes is just the same, since every test comprising of a scalar product can be extended this way. In addition to the incidence relation the notation for the relations parallelism ($\|$, $\|^-$ and $\|^+$) and orthogonality ($\perp$, $\perp^-$ and $\perp^+$) are introduced as well in the case of scalar valued test statistics. If the test statistic instead is vector valued and bilinear, the situation is a little more involved. Let us first consider the case of a point $\mathbf{X}$ and a line $\mathbf{L}$ in 3D-space. According to [5], there is no reason to reject the hypothesis $\mathbf{X} \in \mathbf{L}$ if
$$\mathbf{d}^T\mathbf{\Sigma}^+_{\mathbf{dd}}\mathbf{d} < T_\alpha \tag{3}$$
with
$$\mathbf{d} = \overline{\mathbf{\Gamma}}^T(\mathbf{L})\mathbf{X} \quad \text{and} \quad \mathbf{\Sigma_{dd}} = \overline{\mathbf{\Pi}}^T(\mathbf{X})\mathbf{\Sigma_{LL}}\overline{\mathbf{\Pi}}(\mathbf{X}) + \overline{\mathbf{\Gamma}}^T(\mathbf{L})\mathbf{\Sigma_{XX}}\overline{\mathbf{\Gamma}}(\mathbf{L})$$
and $T_\alpha$ chosen according to the $\chi^2_2$-distribution (see [5] for the definition of the matrices $\mathbf{\Pi}$ and $\mathbf{\Gamma}$). Since $\mathbf{d}$ is a vector, the notion of a single sign is not applicable here. However a test can be formulated, whether two points $\mathbf{X}$ and $\mathbf{Y}$ lie on opposite sides of $\mathbf{L}$, by requiring, that $\mathbf{X}$ and $\mathbf{Y}$ lie on opposite sides of each

of the four planes defined by the rows of $\overline{\mathbf{\Gamma}}^T(\mathbf{L})$, that is, if $\overline{\mathbf{\Gamma}}^T(\mathbf{L})\mathbf{X} = -\overline{\mathbf{\Gamma}}^T(\mathbf{L})\mathbf{Y}$. Thus one obtains the following statistical test: if for all $i = 1..4$ the condition $\left(\frac{d_i^x}{\sigma_{d_i^x}} < T_\alpha \wedge \frac{d_i^y}{\sigma_{d_i^y}} > -T_\alpha\right) \vee \left(\frac{d_i^y}{\sigma_{d_i^y}} < T_\alpha \wedge \frac{d_i^x}{\sigma_{d_i^x}} > -T_\alpha\right)$ with

$$d_i^x = \overline{\mathbf{\Gamma}_i}^T(\mathbf{L})\mathbf{X} \qquad \sigma_{d_i^x}^2 = \overline{\mathbf{\Pi}_i}^T(\mathbf{X})\mathbf{\Sigma_{LL}}\overline{\mathbf{\Pi}_i}(\mathbf{X}) + \overline{\mathbf{\Gamma}_i}^T(\mathbf{L})\mathbf{\Sigma_{XX}}\overline{\mathbf{\Gamma}_i}(\mathbf{L})$$

$$d_i^y = \overline{\mathbf{\Gamma}_i}^T(\mathbf{L})\mathbf{Y} \qquad \sigma_{d_i^y}^2 = \overline{\mathbf{\Pi}_i}^T(\mathbf{Y})\mathbf{\Sigma_{LL}}\overline{\mathbf{\Pi}_i}(\mathbf{Y}) + \overline{\mathbf{\Gamma}_i}^T(\mathbf{L})\mathbf{\Sigma_{YY}}\overline{\mathbf{\Gamma}_i}(\mathbf{L})$$

and $T_\alpha$ chosen according to the $\chi^2$-distribution holds, then there is no reason to reject the hypothesis, that $\mathbf{X}$ and $\mathbf{Y}$ lie on opposite sides of $\mathbf{L}$. This will be denoted by $(\mathbf{X}, \mathbf{Y}) \in^{\otimes} \mathbf{L}$. Every bilinear test statistic can be used this way, although the interpretations of the test are not as clear as in the case of point-line incidence.

### 2.2 Representing Line Segments and their Tests

First consider the 2D-case again: A line segment can be represented by its two end-points $\mathbf{x}$ and $\mathbf{y}$, the line $\mathbf{l}$ connecting those two end-points and the two lines $\mathbf{m}$ and $\mathbf{n}$, orthogonally intersecting $\mathbf{l}$ in $\mathbf{x}$ and $\mathbf{y}$ and directed, such that their normals point away from the line segment. More details about the construction of such line segments can be found in [2].

Again the construction generalizes to 3D line segments in a straightforward manner, by using the end-points $\mathbf{X}$ and $\mathbf{Y}$, the connecting line $\mathbf{L}$ and the planes $\mathbf{A}$ and $\mathbf{B}$ orthogonally intersecting $\mathbf{L}$ in the points $\mathbf{X}$ and $\mathbf{Y}$, directed, such that their normals point away from the line segment.

It is now possible to perform a sequence of statistical tests on the base elements to obtain a result for the compound entity. For example the incidence of a 2D point $\mathbf{z}$ with the 2D line segment $(\mathbf{x}, \mathbf{y}, \mathbf{l}, \mathbf{m}, \mathbf{n})$ can be defined as either $\mathbf{z}$ being incident to one of the endpoints $\mathbf{x}$ or $\mathbf{y}$, or $\mathbf{z}$ being incident to the connecting line $\mathbf{l}$ and lying between the two directed lines $\mathbf{m}$ and $\mathbf{n}$. In the previous notation with logical *and* denoted by $\wedge$ and logical *or* denoted by $\vee$ this then looks like: $\mathbf{z} \equiv \mathbf{x} \vee \mathbf{z} \equiv \mathbf{y} \vee (\mathbf{z} \in \mathbf{l} \wedge \mathbf{z} \in^- \mathbf{m} \wedge \mathbf{z} \in^- \mathbf{n})$. Other statistical tests including incidence, equality, orthogonality and parallelity with 2D line segments are derived easily in a similar manner (cf. [2] for details). In case of 3D line segments some useful relations are summarized in table 1. It can be seen, that a lot of useful statistical tests can be performed very easily with the proposed representation for line segments.

## 3 Storing Uncertain Geometric Entities

### 3.1 Necessary Conditions

Now a data structure will be developed, that allows to efficiently find all uncertain entities, that match a given bilinear relation with a given uncertain entity,

| Entity | Relation | Tests |
|---|---|---|
| point $\mathbf{Z}$ | incident | $(\mathbf{Z} \equiv \mathbf{X}_1) \vee (\mathbf{Z} \equiv \mathbf{Y}_1) \vee ((\mathbf{Z} \in \mathbf{L}_1) \wedge (\mathbf{Z} \in^- \mathbf{A}_1) \wedge (\mathbf{Z} \in^- \mathbf{B}_1))$ |
| line $\mathbf{M}$ | intersect | $\mathbf{L}_1 \in \mathbf{M} \wedge (\mathbf{X}_1, \mathbf{Y}_1) \in^{\otimes} \mathbf{M}$ |
| | orthogonal | $\mathbf{L}_1 \in \mathbf{M} \wedge (\mathbf{X}_1, \mathbf{Y}_1) \in^{\otimes} \mathbf{M} \wedge \mathbf{L}_1 \perp \mathbf{M}$ |
| | parallel | $\mathbf{L}_1 \parallel \mathbf{M}$ |
| | incident | $\mathbf{L}_1 \equiv \mathbf{M}$ |
| plane $\mathbf{C}$ | intersect | $(\mathbf{X}_1, \mathbf{Y}_1) \in^{\otimes} \mathbf{C}$ |
| | incident | $\mathbf{L}_1 \in \mathbf{C}$ |
| | orthogonal | $(\mathbf{X}_1, \mathbf{Y}_1) \in^{\otimes} \mathbf{C} \wedge \mathbf{L}_1 \perp \mathbf{C}$ |
| | parallel | $\mathbf{L}_1 \parallel \mathbf{C}$ |
| line segment $(\mathbf{X}_2, \mathbf{Y}_2, \mathbf{L}_2, \mathbf{A}_2, \mathbf{B}_2)$ | intersect | $\mathbf{L}_1 \in \mathbf{L}_2 \wedge (\mathbf{X}_1, \mathbf{Y}_1) \in^{\otimes} \mathbf{L}_2 \wedge (\mathbf{X}_2, \mathbf{Y}_2) \in^{\otimes} \mathbf{L}_1$ |
| | orthogonal | $\mathbf{L}_1 \in \mathbf{L}_2 \wedge (\mathbf{X}_1, \mathbf{Y}_1) \in^{\otimes} \mathbf{L}_2 \wedge (\mathbf{X}_2, \mathbf{Y}_2) \in^{\otimes} \mathbf{L}_1 \wedge \mathbf{L}_1 \perp \mathbf{L}_2$ |
| | parallel | $((\mathbf{X}_1 \in^- \mathbf{A}_2 \wedge \mathbf{Y}_1 \in^- \mathbf{B}_2) \vee (\mathbf{X}_1 \in^- \mathbf{B}_2 \wedge \mathbf{Y}_1 \in^- \mathbf{A}_2)) \wedge \mathbf{L}_1 \parallel \mathbf{L}_2$ |
| | incident | $((\mathbf{X}_1 \in^- \mathbf{A}_2 \wedge \mathbf{Y}_1 \in^- \mathbf{B}_2) \vee (\mathbf{X}_1 \in^- \mathbf{B}_2 \wedge \mathbf{Y}_1 \in^- \mathbf{A}_2)) \wedge \mathbf{L}_1 \equiv \mathbf{L}_2$ |
| | equal | $(\mathbf{X}_1 \equiv \mathbf{X}_2 \wedge \mathbf{Y}_1 \equiv \mathbf{Y}_2) \vee (\mathbf{X}_1 \equiv \mathbf{Y}_2 \wedge \mathbf{Y}_1 \equiv \mathbf{X}_2)$ |

**Table 1.** Relations with the 3D line segment $(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{L}_1, \mathbf{A}_1, \mathbf{B}_1)$

e.g. given a line segment, one is able to find all those line segments, that orthogonally intersect the given one, out of a large set of stored line segments. Therefore a necessary condition for bilinear tests like eq. (3) is derived first. The generic bilinear test has the form

$$\mathbf{d}^T \mathbf{\Sigma}_{\mathbf{dd}}^{-1} \mathbf{d} < T_{\alpha,n} \tag{4}$$

with

$$\mathbf{d} = \mathsf{A}(\mathbf{x})\mathbf{y} \quad \text{and} \quad \mathbf{\Sigma}_{\mathbf{dd}} = \mathsf{A}(\mathbf{x})\mathbf{\Sigma}_{\mathbf{yy}}\mathsf{A}(\mathbf{x})^T + \mathsf{B}(\mathbf{y})\mathbf{\Sigma}_{\mathbf{xx}}\mathsf{B}(\mathbf{y})^T$$

With $\sigma_x^2$ denoting the largest eigenvalue of $\mathbf{\Sigma}_{\mathbf{xx}}$, $\sigma_y^2$ denoting the largest eigenvalue of $\mathbf{\Sigma}_{\mathbf{yy}}$ and the rows of $\mathsf{A}$ and $\mathsf{B}$ denoted by $\mathbf{a}_i$ and $\mathbf{b}_i$, a necessary condition for eq. (4) is given by

$$\frac{(\mathbf{a}_1^T \mathbf{y})^2}{\sigma_y^2 \mathbf{a}_1^T \mathbf{a}_1 + \sigma_x^2 \mathbf{b}_1^T \mathbf{b}_1} + \cdots + \frac{(\mathbf{a}_n^T \mathbf{y})^2}{\sigma_y^2 \mathbf{a}_n^T \mathbf{a}_n + \sigma_x^2 \mathbf{b}_n^T \mathbf{b}_n} < T_{\alpha,n}$$

Since all terms are positive, this can only hold, if

$$\begin{aligned} &\forall i \quad \frac{(\mathbf{a}_i^T \mathbf{y})^2}{\sigma_y^2 \mathbf{a}_i^T \mathbf{a}_i + \sigma_x^2 \mathbf{b}_i^T \mathbf{b}_i} < T_{\alpha,n} \\ \Leftrightarrow &\forall i \quad \frac{|\mathbf{a}_i^T \mathbf{y}|}{|\mathbf{a}_i||\mathbf{y}|} < \sqrt{T_{\alpha,n} \left( \frac{\sigma_y^2}{\mathbf{y}^T \mathbf{y}} + \frac{\sigma_x^2}{\mathbf{a}_i^T \mathbf{a}_i} \frac{\mathbf{b}_i^T \mathbf{b}_i}{\mathbf{y}^T \mathbf{y}} \right)} \overset{(1)}{<} \sqrt{T_{\alpha,n}} \frac{\sigma_y}{|\mathbf{y}|} + \sqrt{T_{\alpha,n}} \frac{\sigma_x}{|\mathbf{a}_i|} \end{aligned}$$

where the inequality (1) holds, because the $\mathbf{b}_i$ are projections of $\mathbf{y}$ onto some subspace for every relation considered (cf. [5]). One can also assume, that all $\boldsymbol{a}_i$ and $\boldsymbol{y}$ are spherically normalized, because the entities in oriented projective space are represented by homogeneous vectors. If one substitutes $\delta_x = \frac{2\sqrt{3}}{3}\sqrt{T_{\alpha,n}}\sigma_x$ and $\delta_y = \frac{2\sqrt{3}}{3}\sqrt{T_{\alpha,n}}\sigma_y$ a necessary condition for eq. (4) (cf. [2] for a proof) is given by

$$|\boldsymbol{a}_i^T \boldsymbol{y}| \leq \begin{cases} \cos\left(\frac{\pi}{2} - \arccos\delta_x - \arccos\delta_y\right) & \text{if } \delta_x + \delta_y \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

This equation has a simple geometric interpretation: The hypothesis test of eq. (4) can only result in not rejecting the hypothesis, if there is a vector $\boldsymbol{a}'$ within the cone with axis $\boldsymbol{a}_i$ and opening angle $\arccos \delta_x$ and another vector $\boldsymbol{y}'$ within the cone with axis $\boldsymbol{y}$ and opening angle $\arccos \delta_y$, so that the vectors $\boldsymbol{a}'$ and $\boldsymbol{y}'$ are perpendicular.

Notice, that reasoning along the same lines yields necessary conditions for the positive and negative orientation test (cf. eq. (2) and eq. (1)):

$$\pm \boldsymbol{a}_i^T \boldsymbol{y} \leq \begin{cases} \cos\left(\frac{\pi}{2} - \arccos \delta_x - \arccos \delta_y\right) & \text{if } \delta_x + \delta_y \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

Thus, associating a key $(\boldsymbol{y}, \delta_y)$ with each base entity $(\mathbf{y}, \boldsymbol{\Sigma}_{\mathbf{yy}})$, one is able to check only using this key together with eq. (5) or (6), if a statistical hypothesis test with the associated entity might result in not rejecting the hypothesis.

### 3.2 Combination of Keys

The next step is to combine two keys $(\boldsymbol{y}_1, \delta_{y_1})$ and $(\boldsymbol{y}_2, \delta_{y_2})$ into a new superkey $(\boldsymbol{y}', \delta_{y'})$, such that the superkey yields a necessary condition for both of the keys. Since all keys represent hypercones, one looks for the enclosing hypercone to calculate the superkey. Note first, that the axis of the superkey's hypercone must lie in the hyperplane spanned by the two axes $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, thus one can first calculate the intersection of the hypercone $(\boldsymbol{y}_1, \delta_{y_1})$ with this hyperplane. Because it lies in the hyperplane, it can be parametrized by $\mathbf{y}_1' = (1-\lambda)\boldsymbol{y}_1 + \lambda \boldsymbol{y}_2$ and inserting into the hypercone condition results in $\left(\frac{\mathbf{y}_1'^T \boldsymbol{y}_1}{|\mathbf{y}_1'|}\right)^2 = 1 - \delta_{y_1}^2$. Solving this quadratic equation for $\lambda$ yields two solutions and thus two vectors $\mathbf{y}_{11}'$ and $\mathbf{y}_{12}'$. Doing the same for the hypercone $(\boldsymbol{y}_2, \delta_{y_2})$ yields two more solutions $\mathbf{y}_{21}'$ and $\mathbf{y}_{22}'$. Two of those four lines must lie on the surface of the enclosing cone, namely those two with the greatest enclosing angle. To find those, one must first orient the lines to point into the same direction as the corresponding hypercone axis. This can simply be achieved by checking signs of scalar products. Together with a spherical normalization one obtains $\boldsymbol{y}_{ij}^* = \text{sign}\left(\mathbf{y}_{ij}^T \mathbf{y}_i\right) \frac{\mathbf{y}_{ij}}{|\mathbf{y}_{ij}|}$. Since the surface of the enclosing hypercone must lie on both sides of the axis $\boldsymbol{y}_2$ in relation to $\boldsymbol{y}_1$, one now determines, which lines lie on which side:

$$\boldsymbol{y}_i^N = \begin{cases} \boldsymbol{y}_{i1}^* & \text{if } \boldsymbol{y}_{3-i}{}^T \boldsymbol{y}_{i1}^* > \boldsymbol{y}_{3-i}{}^T \boldsymbol{y}_{i2}^* \\ \boldsymbol{y}_{i2}^* & \text{otherwise} \end{cases} \qquad \boldsymbol{y}_i^F = \begin{cases} \boldsymbol{y}_{i1}^* & \text{if } \boldsymbol{y}_{3-i}{}^T \boldsymbol{y}_{i1}^* < \boldsymbol{y}_{3-i}{}^T \boldsymbol{y}_{i2}^* \\ \boldsymbol{y}_{i2}^* & \text{otherwise} \end{cases}$$

Finally one is able to select those two oriented lines, that include both hypercones, again by simply checking scalar products:

$$\boldsymbol{m} = \begin{cases} \boldsymbol{y}_1^F & \text{if } \boldsymbol{y}_2{}^T \boldsymbol{y}_1^F < \boldsymbol{y}_2{}^T \boldsymbol{y}_2^N \\ \boldsymbol{y}_2^N & \text{otherwise} \end{cases} \qquad \boldsymbol{n} = \begin{cases} \boldsymbol{y}_2^F & \text{if } \boldsymbol{y}_1{}^T \boldsymbol{y}_2^F < \boldsymbol{y}_1{}^T \boldsymbol{y}_1^N \\ \boldsymbol{y}_1^N & \text{otherwise} \end{cases}$$

Thus the superkey is now given by:

$$\boldsymbol{y}' = \frac{\boldsymbol{m} + \boldsymbol{n}}{|\boldsymbol{m} + \boldsymbol{n}|}$$

$$\delta_{y'} = \begin{cases} \sqrt{1-(\boldsymbol{m}^T\boldsymbol{y}')^2} & \text{if } \boldsymbol{y}'^T\boldsymbol{y}_1^* > 0 \wedge \boldsymbol{y}'^T\boldsymbol{y}_2^* > 0 \\ 1 & \text{otherwise} \end{cases}$$

By definition it has the property, that whenever eq. (5), or (6) holds for any of $(\boldsymbol{y}_1, \delta_{y_1})$ or $(\boldsymbol{y}_2, \delta_{y_2})$, it must hold for $(\boldsymbol{y}', \delta_{y'})$. Also notice, that it can easily be generalized to more than two keys just by sequentially enlarging the hypercone.

### 3.3 The Data Structure

Having defined those keys, one is now able to define an R-Tree like data structure (cf. [6]), that allows to store compound uncertain geometric entities of a single type, as follows:

- every node of the tree contains at most $2M$ and at least $M$ elements, unless it is the root
- the elements of the leaf nodes are the compound uncertain geometric entities $((\mathbf{y}_1, \boldsymbol{\Sigma}_{\mathbf{y}_1\mathbf{y}_1}), ..., (\mathbf{y}_n, \boldsymbol{\Sigma}_{\mathbf{y}_n\mathbf{y}_n}))$ together with a key $((\boldsymbol{y}_1, \delta_{y_1}), ..., (\boldsymbol{y}_n, \delta_{y_n}))$ as defined in section 3.1
- every inner node's link is associated with a key $((\boldsymbol{y}'_1, \delta_{y'_1}), ..., (\boldsymbol{y}'_n, \delta_{y'_n}))$ constructed from the subnode's keys as described in section 3.2

Two facts follow immediately from the definition of the tree: first its height is bounded by $O(\log N)$ (cf. [1]) and second a statistical test with an entity stored in a leaf node can only result in not rejecting the hypothesis, if for all keys along the path to the root, the eq. 5 or 6 (depending on the test) holds. The second property is used to define the query algorithm for the data structure, by only descending into a subtree if the necessary condition with the key holds. Thus, the more complex the query, the better is the performance of the algorithm, since more subtrees can be truncated at an earlier point in time.

To insert an element into the tree and maintain the first property, a strategy similar to the construction of an R-Tree is used. On every level the algorithm computes for every subtree the enlargement of the opening angles of the keys hypercones and inserts the entity into the subtree, where the enlargements are minimal. If a node has more than $2M$ elements, the node is split into two subsets of size $M$ and $M+1$, such that the opening angles of the superkeys hypercones of the elements of each subset are minimal. To find those subsets, the quadratic split heuristic proposed in [6] is used. As shown in [1], the running time of this algorithm is bounded by $O(logN)$.

A more detailed description together with some implementation details can be found in [2]. Note also, that the data structure is not limited to line segments, but can store and perform any kind of statistical test on data, that is constructed from multiple (or single) uncertain base entities of the Grassmann-Cayley algebra. For points it is similar to the classical R-Tree, thus a similar performance can be expected in this case.

## 4 Experimental Evaluation

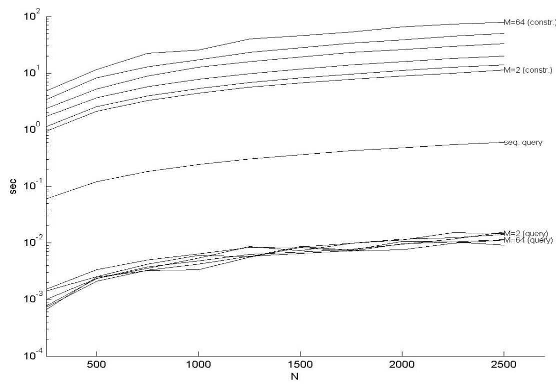


**Fig. 1.** Running times for construction and intersection queries of 3D line segments

The running times of the data structure proposed in the previous section were evaluated on artificial line segment data in 3D-space. A set of $N$ line segments inside the cube of volume 1 centered at the origin were generated randomly. The line segments were of random length between 0.05 and 0.1 and random orientation with the standard deviation of the endpoints being 0.001. All $N$ line segments were inserted into the proposed data structure and another random line segment was used to retrieve all line segments from the data structure, that intersect the given one. Intersection was chosen, because it has the broadest field of application, though the other relations behave very similar. An application example for this kind of query that benefits from the proposed data structure can be found in [3]. The running times on a current standard desktop computer for different values of the nodes half size $M$ are depicted at the bottom of figure 1.

Since classical multi-dimensional data structures do not support statistical geometric tests as query, the running time for sequentially comparing all $N$ line segments with the given one is shown in the middle of figure 1. It can be seen, that the improvement is up to a factor of 50, depending on the number of line segments stored in the data structure.

The drawback of using an index structure is, that the construction requires time. The construction times for different values of $M$ are depicted on top of figure 1. It can be seen, that the choice of $M = 2$ is best, since the construction time heavily depends on $M$ and the query times do not depend on $M$ so strongly. Certainly a large amount of queries, for example required by a spatial join algorithm, to a fairly large and static set of line segments is required to exploit the benefits of the proposed data structure.

## 5 Conclusion

In this work a framework was presented, that allows to perform statistical tests on uncertain geometric entities constructed from tuples of uncertain base entities in oriented projective space. It was shown, that uncertain line segments in 2D- and 3D-space are constructible in such a way, that statistical reasoning about this practically very important geometric entities is possible in this framework.

The second contribution of this work is the introduction of a data structure, that allows to perform this kind of tests in an efficient manner. The special structure of statistical testing was used in the design of the data structure, such that it is capable of performing complex statistical reasoning tasks in an efficient manner. It therefore outperforms classical multi-dimensional data structures, since they are not able to handle this kind of queries.

Since the amount of measured, i.e. uncertain, geometric data in many computer vision tasks is extremely high, the need for efficient geometric reasoning algorithms is evident. The experiments showed, that the gain in performance is very high, if large amounts of data are to be processed, so that the application of the presented framework and data structure could lead to new, more feasible algorithms in the analysis of large aerial images or large image sequences, where known statistical properties of the measured data can be used.

## References


1. R. Bayer and E. McCreight. Organization and maintenance of large ordered indexes. *Acta Informatica*, 1:173–189, 1972.
2. C. Beder. Joinalgorithmus für Mengen unsicherer geometrischer Elemente. Technical report, Institut für Photogrammetrie, 2003.
3. C. Beder. A unified framework for the automatic matching of points and lines in multiple oriented images. In *Proc. 20th ISPRS Congress, Istanbul, Turkey*, 2004.
4. R. Finkel and J. Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974.
5. Brunn. A. Förstner, W. and S. Heuel. Statistically testing uncertain geometric relations. In *Proc. DAGM 2000, Kiel, Germany*, pages 17–26, 2000.
6. A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 47–57, 1984.
7. 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.
8. S. Heuel and W. Förstner. Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. In *CVPR '01*. IEEE, 2001.
9. F. Jung and N. Paparoditis. Extracting 3d free-form surface boundaries of man-made objects from multiple calibrated images: A robust, accurate and high resolving power edgel matching and chaining approach. In *Proc. of the ISPRS Conf. Photogrammetric Image Analysis*, pages 39–44, 2003.
10. D. E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, 1998.
11. David G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.
12. R. Schneider R. Seeger N. Beckmann, H. Kriegel. The R*-tree: An efficient and robust access method for points and rectangles. In *Proc. ACM SIGMOD Symposium on Principles of Database Systems*, pages 322–331, 1990.
13. J. Stolfi. *Oriented Projective Geometry*. Academic Press, 1991.