Institut für Geodäsie und Geoinformation
Bereich Photogrammetrie

D 98

Grouping Uncertain Oriented Projective Geometric Entities with Application to Automatic Building Reconstruction

Inaugural-Dissertation
zur
Erlangung des Grades
Doktor-Ingenieur
(Dr.-Ing.)
der
Hohen Landwirtschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität
zu Bonn

vorgelegt am 30. Oktober 2006 von
Christian Beder
aus Bonn

Referent:	Prof. Dr. Wolfgang Förstner
Korreferent:	Prof. Dr. Hans-Peter Helfrich
Tag der mündlichen Prüfung:	14.12.2006
Erscheinungsjahr:	2007
Gedruckt bei:	Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn `http://hss.ulb.uni-bonn.de/diss_online` elektronisch publiziert.

Zusammenfassung

Gruppierung unsicherer orientierter projektiver geometrischer Elemente mit Anwendung in der automatischen Gebäuderekonstruktion

Die vollautomatische Rekonstruktion von 3D Szenen aus einer Menge von 2D Bildern war immer ein Hauptanliegen in der Photogrammetrie und Computer Vision und wurde bisher noch nicht zufriedenstellend gelöst. Die meisten aktuellen Ansätze ordnen Merkmale zwischen den Bildern basierend auf radiometrischen Eigenschaften zu. Daran schließt sich dann eine Rekonstruktion auf der Basis der Bildgeometrie an. Die Motivation für diese Arbeit ist die These, dass es möglich sein sollte, die Struktur einer Szene durch Gruppierung geometrischer Primitive zu rekonstruieren, falls die Eingabedaten genügend redundant sind.

Orientierte projektive Geometrie wird in dieser Arbeit zur Repräsentation geometrischer Primitive, wie Punkten, Linien und Ebenen in 2D und 3D sowie projektiver Kameras, zusammen mit ihrer Unsicherheit verwendet.

Der erste Hauptbeitrag dieser Arbeit ist die Verwendung unsicherer *orientierter* projektiver Geometrie, anstatt von unsicherer projektiver Geometrie, welche die Repräsentation von komplexeren *zusammengesetzten* Objekten, wie Liniensegmenten und Polygonen in 2D und 3D sowie 2D Edgels und 3D Facetten, ermöglicht. Innerhalb dieser unsicheren orientierten projektiven Repräsentation wird ein Verfahren zum testen paarweiser Relationen zwischen den verschiedenen unsicheren orientierten projektiven geometrischen Elementen entwickelt. Dabei liegt die Neuheit wieder in der Möglichkeit, Relationen zwischen den neuen zusammengesetzten Elementen zu prüfen.

Der zweite Hauptbeitrag dieser Arbeit ist die Entwicklung einer Datenstruktur, welche speziell auf die effiziente Prüfung von solchen Relationen zwischen vielen Elementen ausgelegt ist. Die Möglichkeit zur effizienten Prüfung von Relationen zwischen den geometrischen Elementen erlaubt nun die Entwicklung eines Systems zur Gruppierung dieser Elemente. Verschiedene Gruppierungsmethoden werden vorgestellt.

Der dritte Hauptbeitrag dieser Arbeit ist die Entwicklung einer neuen Gruppierungsmethode, die durch die Analyse der Änderung der Entropie beim Hinzufügen von Beobachtungen in die Schätzung Effizienz und Robustheit gegeneinander ausbalanciert und dadurch bessere Gruppierungsergebnisse erzielt.

Zum Schluss wird die Anwendbarkeit der vorgeschlagenen Repräsentationen, Tests und Gruppierungsmethoden für die ausschließlich geometriebasierte Gebäuderekonstruktion aus orientierten Luftbildern demonstriert. Es wird gezeigt, dass unter der Annahme von hoch redundanten Datensätzen vernünftige Rekonstruktionsergebnisse durch Gruppierung von geometrischen Primitiven erzielbar sind.

Abstract

Grouping Uncertain Oriented Projective Geometric Entities with Application to Automatic Building Reconstruction

The fully automatic reconstruction of 3d scenes from a set of 2d images has always been a key issue in photogrammetry and computer vision and has not been solved satisfactory so far. Most of the current approaches match features between the images based on radiometric cues followed by a reconstruction using the image geometry. The motivation for this work is the conjecture that in the presence of highly redundant data it should be possible to recover the scene structure by grouping together geometric primitives in a bottom-up manner.

Oriented projective geometry will be used throughout this work, which allows to represent geometric primitives, such as points, lines and planes in 2d and 3d space as well as projective cameras, together with their uncertainty.

The first major contribution of the work is the use of uncertain *oriented* projective geometry, rather than uncertain projective geometry, that enables the representation of more complex *compound* entities, such as line segments and polygons in 2d and 3d space as well as 2d edgels and 3d facets. Within the uncertain oriented projective framework a procedure is developed, which allows to test pairwise relations between the various uncertain oriented projective entities. Again, the novelty lies in the possibility to check relations between the novel compound entities.

The second major contribution of the work is the development of a data structure, specifically designed to enable performing the tests between large numbers of entities in an efficient manner. Being able to efficiently test relations between the geometric entities, a framework for grouping those entities together is developed. Various different grouping methods are discussed.

The third major contribution of this work is the development of a novel grouping method that by analyzing the entropy change incurred by incrementally adding observations into an estimation is able to balance efficiency against robustness in order to achieve better grouping results.

Finally the applicability of the proposed representations, tests and grouping methods for the task of purely geometry based building reconstruction from oriented aerial images is demonstrated. It will be shown that in the presence of highly redundant datasets it is possible to achieve reasonable reconstruction results by grouping together geometric primitives.

Contents

1 Introduction **7**
- 1.1 Motivation . . . 7
- 1.2 Previous work on fundamental theory . . . 8
  - 1.2.1 Previous work on uncertainty . . . 9
  - 1.2.2 Previous work on projective geometry . . . 15
  - 1.2.3 Previous work on data structures and algorithms . . . 17
  - 1.2.4 Previous work on grouping . . . 19
- 1.3 Previous work on required pre-processing . . . 25
  - 1.3.1 Previous work on calibration . . . 25
  - 1.3.2 Previous work on structure from motion . . . 26
  - 1.3.3 Previous work on feature extraction . . . 32
- 1.4 Previous work on competing techniques . . . 36
  - 1.4.1 Previous work on feature matching . . . 37
  - 1.4.2 Previous work on building reconstruction . . . 45

2 Uncertain oriented projective geometry **51**
- 2.1 Representing uncertain oriented entities . . . 51
  - 2.1.1 Uncertain oriented 2d base entities . . . 54
  - 2.1.2 Uncertain oriented 2d compound entities . . . 57
  - 2.1.3 Uncertain oriented 3d base entities . . . 62
  - 2.1.4 Uncertain oriented 3d compound entities . . . 68
  - 2.1.5 Uncertain oriented projective cameras . . . 80
- 2.2 Testing geometric relations . . . 89
  - 2.2.1 Relations between the base entities . . . 91
  - 2.2.2 Relations between the compound entities . . . 99
- 2.3 Data structures for efficient testing . . . 120
  - 2.3.1 Necessary conditions . . . 120
  - 2.3.2 Combining necessary conditions . . . 127
  - 2.3.3 The tree data structure . . . 130
  - 2.3.4 Empirical performance analysis . . . 137

3 Grouping **143**
- 3.1 Motivation . . . 143
- 3.2 Problem definition . . . 144
- 3.3 Geometric grouping tasks . . . 146
  - 3.3.1 Grouping into 2d points . . . 148
  - 3.3.2 Grouping into 2d lines . . . 149
  - 3.3.3 Grouping into 3d points . . . 150

3.3.4 Grouping into 3d lines . . . 151
3.3.5 Grouping into 3d planes . . . 155
3.4 Grouping methods . . . 156
3.4.1 Expectation maximization . . . 157
3.4.2 Graph based approach . . . 157
3.4.3 Random Sample Consensus method . . . 160
3.4.4 Entropy bound approach . . . 161
3.5 Empirical analysis on synthetic data . . . 170

4 Application to building reconstruction 175
4.1 Input data set and task . . . 175
4.2 Grouping results . . . 175

5 Conclusion 193

Chapter 1

Introduction

1.1 Motivation

The description of a depicted scene from images has always been the ultimate goal of photogrammetry and computer vision. One important aspect of this task is the three dimensional reconstruction of the visible world. Though a lot of progress has been achieved in recent years, the problem of fully automatic reconstruction without human interaction is still unsolved for a variety of important reconstruction tasks including the fully automatic building reconstruction from aerial images that has always received a lot of attention from the photogrammetric community.

The availability of digital cameras that has taken place in recent years has decreased the expenses for acquiring large amounts of images significantly. Hence, highly redundant sets of high quality images have become available at very low cost and the potential impact of this redundancy has only been recognized for the goal of full automation by few people in the photogrammetric community so far (cf. Gruber *et al.* [2003]).

In addition to this more practical aspects, the research on the theory underlying the geometry of multiple images has been very active in the last decade and much progress has been achieved. Especially the work of Heuel [2004], who proposed a methodology for reasoning about geometric entities in the presence of uncertainty, has influenced this work very strongly.

The main motivation for this work is the conjecture that in the presence of highly redundant data it should be possible to recover the complete scene structure by grouping together geometric primitives without using the radiometric image content (see figure 1.1). It will be exploited, how far a scene can be reconstructed by grouping together geometric primitives in a bottom-up manner. Neither high-level prior knowledge nor radiometric information from the images content should be integrated into this methodology, in order to show how much the purely geometric information contributes to the results. It will be shown that reasonable results are achievable by this purely geometric approach, strengthening the vital importance of integrating this kind of information into every image matching procedure.

This work is structured as follows: First the relevant previous work in the areas of uncertainty, projective geometry, algorithms, grouping, camera calibration, structure-from-motion, feature extraction, feature matching and building reconstruction will be reviewed.

Next, the concepts of uncertain oriented projective geometry will be introduced. Representations for various geometric entities including points, lines, planes, line segments and polygons in 2d and 3d space are presented. It will be shown, how various relations between those uncertain entities can be tested. The novelty lies in the additional relations between

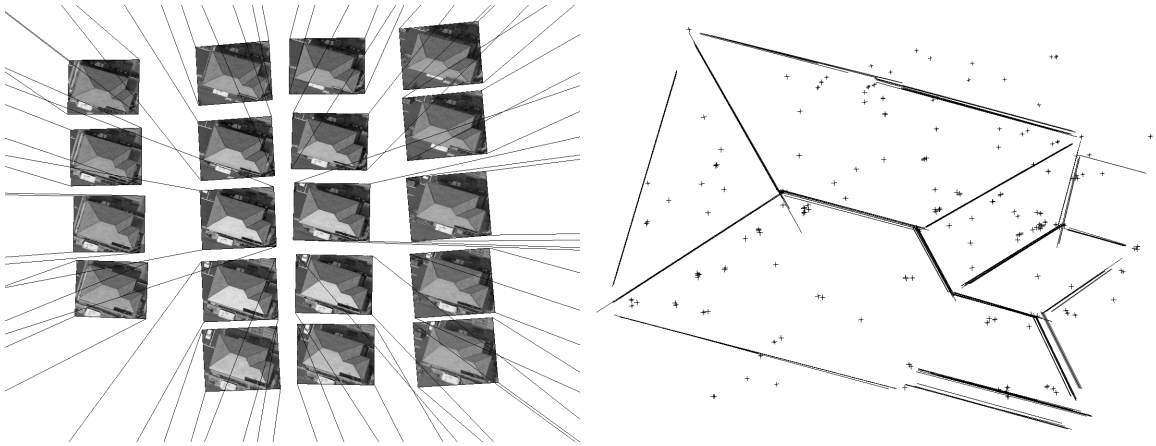

Figure 1.1: *Left:* A highly redundant set of digital aerial images showing a single house. The orientation and calibration of the images are known and indicated by the lines connecting the four image corners with the projection centers. *Right:* The reconstruction obtained by grouping the geometric primitives without using any intensity information from the images for matching. This example, which will be picked up again in the last chapter of this work, shows the importance of geometric (rather than radiometric) cues for feature matching.

the oriented uncertain entities that allow to represent additional compound entities, such as line segments and polygons, in a unified and simple framework. Also a novel data structure will be presented, which allows to efficiently perform those tests and enables the practical application of the presented framework to large real world datasets. An empirical performance assessment of the data structure will demonstrate its applicability and usefulness.

The third part will be concerned with grouping such uncertain oriented projective entities together. The grouping problem will be formally stated and various useful tasks that could be solved as geometric grouping problems will be presented. Four different grouping methodologies for solving the stated grouping problem will be presented. Especially the fourth method based on analyzing the expected entropy increase in an incremental estimation presents a major contribution of this work. The performance of all grouping methods will be compared empirically for a synthetic data set with known ground truth.

Finally the applicability of the grouping methodology for the task of fully automatic building reconstruction, more precisely the reconstruction of 3d points and 3d lines, from oriented aerial images will be demonstrated. The differences between the results of the graph based grouping approach and the entropy bound grouping approach will be discussed.

1.2 Previous work on fundamental theory

In this section some previous work on the fundamental theory underlying the contributions of this work will be presented. First previous work on methods for dealing with uncertain data, namely probability theory and information theory, will be discussed. Then the previous work in the field of projective geometry will be reviewed. Some work on algorithms and data structures related to the contributions of this work will be shown in section 1.2.3. Finally the previous work on grouping will be presented in section 1.2.4.

1.2.1 Previous work on uncertainty

The inherent uncertainty of the data must be considered as a key challenge in computer vision. The rigorous treatment of this uncertainty has therefore received a lot of attention, as it not only improves the results compared to ad-hoc methods but also makes the results more explainable. It will be shown in this work, how the rigorous treatment of uncertainties not only reduces the need for unexplainable thresholds, but also allows to derive new algorithms through its analysis (cf. section 3.4.4).

Mumford [2000] makes the very polemic point that statistical rather than logical reasoning will be even the future of mathematics itself. It is argued that the stringent logical reasoning, which has influenced the thinking in modern science, is not leading to successful results and that the modeling of uncertainty will become the scientific paradigm of the future.

For modeling uncertainty, there exist two important interrelated and supplementary concepts, namely probability theory and information theory. The concepts must be considered supplementary, as information theory is built upon probability theory, but its concepts are extremely useful by themselves. The following two sections discuss some previous work on each of those two subjects.

Probability theory

This section summarizes some previous work on probability theory that will be used in this work to deal with the uncertainty of the measured geometric data. Though probability theory has been recognized as being very useful in recent years, there had been a lot of criticism on the applicability of probability theory for the task of reasoning from the artificial intelligence community. Answering this doubts, Cheeseman [1985] defends the tools provided by probability theory and argues that probability theory is the right tool for dealing with uncertainty. Various criticism on Bayesian statistics from the artificial intelligence community are tackled and it is argued that many forms of reasoning systems are reducible or inferior to the use of probabilistic methods for the same task.

A very good textbook, which first introduced the basic concepts of probability theory to a broader audience in the western world, is [Feller, 1968]. From this the whole field of statistics, which is here seen as the sub-field of probability theory that is concerned with estimating probabilistic models from observations emerged. In the geodetic context uncertainty modeling and estimation techniques have always received a lot of attention with [Mikhail and Ackermann, 1976], [Koch, 1997; 1988] and [Niemeier, 2001] being the most influential text books for this work. Another important book in the field is [Kanatani, 1996], which explicitly is about estimation techniques in the 3d computer vision context.

One particular important aspect of statistics is the estimation of parameters of a known model from uncertain input data. In [Förstner and Wrobel, 2004] a broad overview over the various estimation techniques in the photogrammetric context is given. Of particular importance in this work (cf. section 3.4.4) is the incremental estimation also known as Kalman-filtering. The tutorial [Welch and Bishop, 1995] gives an introduction to discrete Kalman filtering that has received a lot of attention in the computer vision community. Also the extended Kalman filter for non-linear processes is discussed there.

While pure estimation is only concerned with small noise introduced by the measurement process, fully automatic procedures often introduce a large amount of outliers into the data. As the target of this work is the development of such fully automatic methods, outliers in the data have to be dealt with. One aspect of dealing with outliers is known as robust estimation that aims at fitting optimal models to the data even in the presence of such gross errors in the observations that do not result from the measurement process itself but from other uncontrollable sources of error. The book [Huber, 1981] is specifically about such robust estimation techniques.

A very important robust estimation technique in computer vision that is also the basis for the grouping algorithm presented in section 3.4.3 is known as random sample consensus (RANSAC). It was introduced by Fischler and Bolles [1981]. The estimation procedure presented there is as follows: a putative model is estimated from a minimal set of data points and the remaining data points are used to validate this model. The process is iterated and the best fitting model is retained. This procedure could also be used for grouping, as will be shown in section 3.4.3.

Performance is always a vital issue that limits the applicability of the RANSAC grouping method to small toy problems (cf. section 3.4.3). In [Matas and Chum, 2005] a method is proposed to speed up the verification part of a RANSAC algorithm. Using a statistical analysis of sequential decision making, the known percentage of outliers is used to derive an optimal number of verification steps.

As pointed out before, Kalman filtering is at the core of the grouping technique presented in section 3.4.4, where robustness is achieved through analyzing the expected entropy increase of each update. A different path is taken by Vedaldi *et al.* [2005], who propose to increase the robustness of Kalman filtering by applying a RANSAC-like inlier/outlier-selection for each update. Every group of new observations is randomly divided into inlier and outlier and the parameter update is computed. The inlier/outlier configuration that maximizes the posterior probability is selected for the update.

While robust estimation in principle is only concerned with obtaining reliable estimates in the presence of outliers, another important aspect of robust statistics is the detection of outliers in the data itself. Though most robust estimation techniques solve also this outlier detection task, it has a special relevance in the grouping context, because grouping can be seen as partitioning the data subsequently into inliers and outleirs (cf. section 3.4.3).

The analysis of incremental parameter estimation techniques is the basis for the grouping method presented in section 3.4.4 and was inspired by the following techniques for detecting outliers.

In [Förstner, 1983] specifically the effect of undetected errors on estimation and statistical hypothesis testing is analyzed. Multiple possible alternative hypothesis are considered and the notion of probability of an error of third type is introduced, which reflects the fact that the wrong hypothesis is selected among the multiple possible models.

Also in [Förstner, 1987] the effect of gross errors on an estimation depending on the design is analyzed. Redundancy numbers are shown to describe the effect of a particular observation on the result and are therefore very useful to decide, if an observation contributes to the model. This approach is very similar to the information theoretic grouping method presented in section 3.4.4 and yields very similar results.

Förstner [2001b] summarizes methods for analyzing the quality of an estimation system. The quality measures that are all purely design dependent are determinability, controllabil-

ity and locatability. Determinability is the instability of the design with respect to small random errors, which can be analyzed using the expected covariance matrices. A measure for comparing covariance matrices is proposed. Controllability is the possibility to detect errors using hypothesis testing, which can be analyzed using the redundancy numbers of the design as already proposed in [Förstner, 1987]. Locatability is the possibility to distinguish between the influence of different groups of observations on the estimation result, which can be analyzed using the off-diagonal elements of the hat-matrix. Though this latter aspect looks promising for the task of grouping, none of the grouping methods presented in section 3.4 will exploit it, yet.

An application of the previous methods is presented in [Sester and Förstner, 1989], where an algorithm for locating objects in images is discussed, which is based on the analysis of the uncertainty of the image features. Not only the precision of the estimated parameters is considered, but also the sensitivity. It is pointed out that bounding the precision from above is not sufficient but that the sensitivity must be bound from below as well as from above, too. This is because neither useless observations nor leverage observations should be taken into account. This latter statement also holds true for grouping tasks, as has already been noted by Förstner [1990], and will be the key for the information theoretic grouping method presented in section 3.4.4.

The scope of all approaches discussed above is not the representation of the uncertainty itself, although they all implicitly assume the uncertainty to be represented by the first two moments of the probability distributions, which is in principle justifiably by a maximum entropy argument. The suitability of this representation is studied by DeVore [2005], who derived error probabilities for two-class hypothesis tests. It is argued that estimated moments are well suited for decision making. In addition to the Gaussian approach, which uses only the first two moments, another approach using the four parameter family of Johnson distributions is presented, which turns out, in spite of the difficulty to estimated higher order moments, to be better suited for low-dimensional problems. This approach will not be followed here and only the first two moments are used.

Also Förstner *et al.* [2000] use the first two moments of distributions of geometric entities, which will be done extensively here, too. The Grassmann-Cayley algebra on projective geometric entities is connected with error propagation and statistical hypothesis testing. The construction of uncertain points, lines and planes in 2d and 3d and the statistical test between those entities are described that are the basis for the representations presented in section 2.1 and test presented in section 2.2.

In [Heuel, 2004; 2001] a method for the optimal estimation of points, lines and planes is described in addition to the representations and tests presented in [Förstner *et al.*, 2000]. The homogeneous representation of the entities together with their covariances is advocated, since all occurring constructions are bilinear in the entities, enabling a simple and rigorous error propagation. The Gauss-Helmert model in conjunction with initial values obtained from the minimum of the algebraic distance, which can easily be computed using a SVD, is proposed for the estimation task. The analysis of this specific estimation technique is the basis for the grouping method presented in section 3.4.4.

The works [Förstner, 2001a; 2001c] present an alternative method for the optimal estimation of such statistical uncertain geometric entities. The bi-linear relations between the entities together with their covariances are integrated into the formulation of an eigenvector problem, which then needs to be solved iteratively. This approach is not followed here.

Perwass *et al.* [2005] extend the notions of the uncertain projective entities presented in [Förstner *et al.*, 2000] toward circles and spheres using the geometric algebra of conformal space. The resulting linear constraints are applicable for estimation, error propagation and statistical testing in the same framework. Therefore those entities straightforwardly fit into the concepts presented in this work as well, but are not exploited any further so far.

The apparent ease of representing uncertain geometric entities using the first two moments of a distribution in projective space, which is also the approach in this work, comes at a price. As will be discussed in detail in section 2.1, the scale ambiguity in projective space requires to fix a gauge scale in order to achieve uniqueness of the representation. This has been studied by Kanatani and Morris [2001], who discuss the problem of a unique description of uncertainty under varying gauge transformations, i.e. the problem that the covariance matrix changes under different equivalent transformations in over-parameterized estimations. The proposed solution is to normalize the entities to a fixed reference gauge and compare the normalized covariance matrices instead. This has also been studied by Heuel [2004].

Morris *et al.* [1999] discuss the effect of gauge constraints on the uncertainties derived from a bundle adjustment. It is pointed out that using the Moore-Penrose inverse of the normal equation matrix gauge constraints are not required to obtain a solution. It is conjectured that therefore the result gives an unambiguous measure for the solution uncertainty. This idea is not follow here, as the gauge constraints are usually known and should be exploited fully in order to obtain stable results.

Kanatani [2004] discusses the measurement uncertainty of image feature operators and its impact on geometric reasoning and model selection. It is pointed out that geometric inference requires statistical modeling. Furthermore it is shown, how the asymptotic behavior of geometric inference for noise going to zero is related to the asymptotic statistical analysis for sample size going to infinity. This is a very desirable property which justifies the chosen representations of uncertain entities.

Other authors have studied uncertain projective geometric entities as well and should be mentioned here. For instance Criminisi *et al.* [1999] analyze the uncertainty of the 2d homography. It is shown, how point measurement errors affect the uncertainty of the homography and vice versa. By analyzing the second order error propagation it is shown that first order approximations, which are made throughout this whole work, are usually sufficient.

In [Csurka *et al.*, 1997] the uncertainty of the fundamental matrix is addressed. The fundamental matrix is re-parameterized in order to obtain a full-rank covariance matrix, which is then derived using error propagation. In [Zhang, 1998] also several estimation techniques for the fundamental matrix are reviewed.

In [Begelfor and Werman, 2005] it is argued that a degenerate normal distribution of the nine parameters of a planar homography is not an adequate probabilistic model for a transformation. Instead another family of distributions is proposed, which exploits the properties of the transformation group and shows a much better performance in Monte-Carlo simulations as well as object recognition. This path is not followed here.

Thus far previous work on uncertain projective base entities has been presented. The idea of uncertain compound entities (cf. section 2.1) occurs in [Shi, 1998], which proposes a representation of uncertain line segments. The idea is to describe the two end-points with its

mean and covariance matrix assuming a Normal distribution. Each point between the two end-points is now considered as a mixture of those two Gaussians and the confidence region of the line segment is defined as the union over all confidence regions on the segment. In [Shi and Liu, 2000] the approach is extended to allow correlated end-points. In contrast to this, the uncertain line segments are composed of oriented projective entities in section 2.1 allowing easier reasoning and estimation using the methods of Heuel [2004; 2001].

Thus far the previous work on probability theory and statistics, which is concerned with the optimal and robust estimation of parameters, has been presented. The previous work on probabilistic methods, which enable the detection of outliers, was reviewed as well as the work on representing uncertain geometric entities. The work on this uncertainty representations included the study of the impact of projective gauge freedom on the uncertainty as well as previous work on compound entities.

Information theory

In the following some previous work in information theory will be reviewed.

The field of information theory was spawned by the classical paper [Shannon, 1948], which addresses the information content of messages and their compressibility. The central quantity of this theory is the entropy, which captures a notion of degree of randomness that will turn out to be very helpful for grouping decisions in section 3.4.4, where the entropy is the central quantity for decision making as well. Therefore some previous work on information theory will be reviewed in the following.

A very good textbook in the field of information theory is [Cover and Thomas, 1991], which introduces all information theoretic notions required in this work in a very clear manner. Also the statistics book [Kagan *et al.*, 1973] discusses information theoretic quantities to characterize probability distributions.

The role of information theory in the context of photogrammetry has already been identified by Förstner [1989], who sketched the impact of information theoretic quantities on some image analysis techniques. Specifically the extraction and matching of image features is viewed in a unifying information theoretic framework.

The importance of information theory in the grouping context is the possibility to quantify some notion of simplicity. Chater and Vitanyi [2003b] give an overview of the role of simplicity in the whole field of cognitive science. It is pointed out that this notion is a unifying factor between the research on patter recognition, information theory, statistics and psychology.

More specifically toward the grouping problem Chater and Vitanyi [2003a] discuss the connection between psychological generalization, i.e. the fact that two items are perceived as belonging to the same group, and the information theoretic notion of Kolmogorov information distance. As both notions are not computable, the discussion is on a fairly philosophical level, although an exponential functional relation between both is stated.

Being much more specific and devising an actual working algorithm, Rissanen [1985] introduces the concept of minimum description length (MDL), which estimates a model of the data based on a given encoding. The close relation between maximum-likelihood estimation

and minimum-length encoding is exploited. The main advantage of the MDL approach is the possibility to estimate the parameters as well as their number in an unified manner. The MDL principle is also a key concept in the grouping context, as optimal groupings should allow a compact description using a group model for each group and a group membership for each observation. This idea will be captured in detail in the definition of the grouping problem presented in section 3.2.

The problem of optimal model selection, which is of vital importance in any grouping task, is discussed by Torr [2002], who studies the connection between Bayesian model estimation and model selection. A RANSAC-like procedure for MAP-estimation is proposed. Furthermore it is pointed out that classical hypothesis testing is not very well suited for deciding among different models. Instead the use of information theoretic quantities is advocated. In [Torr, 1997] this generic framework is applied to the problem of deciding on a correct motion model between an image pair.

The minimization of description length is also the goal of Davison [2005], who analyzes the effect of a Kalman update on the mutual information. In contrast to [Beder, 2005] (also presented in section 3.4.4) it is proposed to choose measurements, which decrease the entropy most, yielding the largest reduction of description length in each step. This leads fastest to the most accurate estimation in the case that there are no outliers. In the presence of outliers this decrease has to be controlled more carefully, which is one of the major contributions of section 3.4.4.

One shortcoming of the MDL principle is that it is dependent on the actual encoding. Invariance against encoding could be achieved theoretically by minimizing the description length over all possible encodings. This invariant quantity is called Kolmogorov complexity, which is obviously practically not computable. Grünwald and Vitanyi [2003] discuss the relation between the concepts of Kolmogorov complexity and the concepts of information theory. It is pointed out that universal coding, which is the basis for nearly all data compression methods, bridges the gap between the two concepts, meaning that finding the optimal representation for the data and determining the generative process underlying the data is essentially the same. In [Grünwald and Vitányi, 2004] those concepts and their relations are extended to the concepts mutual information and sufficient statistics, both in the classical probabilistic version and in the Kolmogorov algorithmic version.

Also Vitanyi [2004] discusses an algorithmic, in contrast to a probabilistic, approach to model selection. The notion of typicality of data is defined in an algorithmic manner and bounds in terms of Kolmogorov complexity are derived. It is argued that this quantity is closely related to minimum-description-length and also maximum-likelihood-estimation with a finite restriction on the model complexity. This algorithmic path is not taken here, though.

In contrast to the ultimate goal of minimizing the description length robust algorithms should be very careful to do so, as greedy strategies could lead to very bad results. Hence, algorithmic decisions should be designed, such that as few restrictions as possible are imposed on the future. In [Jaynes, 1982] this notion is formalized and the idea behind the maximum-entropy inference is discussed. Therefore a theorem is presented that states that under certain assumptions the majority of probability distributions, which could have generated a finite data set, must have an entropy close to the maximum entropy distribution.

This seeming contradiction between the simultaneous requirement to minimize and to

maximize the entropy can be resolved as follows: Obviously the final MDL result should have minimal entropy amongst all solutions. On the other hand leverage points significantly decrease the entropy, so that greedy optimization strategies get trapped in local minima of the description length. Therefor one has to trade efficiency against robustness, which will be detailed in section 3.4.4, in order to get satisfactory results in the presence of leverage points.

Grünwald and Dawid [2004] relate this maximum entropy approach and the robust Bayesian decision theory in a game theoretic setting. It is shown that maximizing the entropy and minimizing the worst-case expected loss are essentially different approaches for solving the same problem. This shows, how maximum entropy methods and robust estimation, which has been identified as closely coupled with the grouping problem in the previous section, are related.

Also Grünwald and Halpern [2004] argue against the common sense that using all available information is sometimes not the best choice. Some examples are presented, where a non-informative prior in connection with small sample sizes yield worse predictions than just ignoring certain given information. It is pointed out that in the context of maximum entropy inference this phenomenon does usually not occur, because here uninformative data is ignored, too.

Trading efficiency against robustness is also a topic of Brand [1999], who relates the concepts of entropy minimization, free energy minimization and posterior maximization. The prior in the MAP estimation is augmented with a temperature term leading to the maximum entropy approach of deterministic annealing for minimizing the overall entropy, i.e. discovering the structure in a robust and efficient way, which is the goal in every grouping task.

In this section some previous work on information theory has been summarized. The unifying concept of simplicity, quantified by the entropy as a measure, is identified as the basis for the work on minimum description length and optimal model selection. Some work on the algorithmic notion of Kolmogorov complexity, which tries to overcome the major disadvantage of MDL to select a specific encoding, were reviewed. Finally, some previous work on maximum entropy methods, which trade efficiency against robustness in the presence of outliers and leverage points, has been presented.

1.2.2 Previous work on projective geometry

This work is concerned with projective geometric entities, as they present an easy method for dealing with projective transformations, which frequently occur in the imaging process. This has also been realized in the early textbook by Blaschke [1948]. In this section some previous work on projective geometry, which has not already been discussed in the previous section on uncertainty, will be presented. Especially the previous work on oriented projective geometry in the computer vision context, which is subject of this work, will be reviewed.

A textbook that introduced the concepts of projective geometry into the field of computer vision is [Faugeras, 1993]. The book [Hartley and Zisserman, 2000] gives a comprehensive and consolidating overview over the field of projective geometry in computer vision and must be considered today as the most important book in this field. Faugeras and Luong [2001] review the field in their book from a more theoretical point of view and give more details about the underlying Grassmann-Cayley algebra.

An introductory overview on the topic of multi view geometry could now be found in many

computer vision books such as [Heyden and Pollefeys, 2004], as the topic is now considered as one of the fundamental aspects in the field. The geometry of the imaging process also has been a key issue in photogrammetric research for many decades and Mugnier *et al.* [2004] give an introduction to the topic of multiple view geometry for the photogrammetric community. The ideas and much of the notation of this and the related work by Heuel [2004] were very influential to this work.

An interesting unifying theoretical framework of multi-view geometry, which has not received much attention in the textbooks mentioned above, is proposed by Triggs [1995b; 1995a], who proposed the concept of the joint image. The basic idea is to consider the space resulting from stacking all projective image spaces into one single product space. It is shown that all matching constraints between the images are expressible in terms of three dimensional linear subspaces of the joint image space. Those linear subspaces allow to apply the techniques from Grassmann-Cayley algebra and contain the well-known epipolar-, the trifocal- and the quadrifocal-constraints.

This work is mainly concerned with oriented projective geometry, as it allows to represent and reason about a much richer class of entities (cf. section 2.1 and section 2.2). The notion of oriented projective geometry was introduced by Stolfi [1991], who extended the notions of projective geometry toward oriented projective geometry, which allows to check signs and therefore augments the concept with many useful new possibilities as shown in section 2.2.

The implications of orientation on the imaging process were studied by Hartley [1998], who introduced the concept of so-called chirality, which captures the very reasonably idea that all points must lie in front of each camera. The notion is introduced in an oriented projective framework and it is shown that not only a projective but a quasi-affine reconstruction of a scene from a pair of uncalibrated images is possible. This quasi-affine reconstruction has the important property that the position of points with respect to a plane stays invariant, i. e. not only "incidence"relations but also "above"and "below"relations are preserved. How this kind of relations in an uncertain oriented projective framework are testable is a major contribution of this work and will be presented in section 2.2. Also note that the result of [Hartley, 1998] implies the applicability of the uncertain oriented projective framework presented here to uncalibrated images (cf. section 1.3.1 for some limitations of this statement). The same idea has also been published by Pajdla *et al.* [1998], who introduce the notion of oriented projective reconstruction. They describe, how, by enforcing all scene points to be in front of the camera and the plane at infinity not to intersect the convex hull of the reconstructed scene points, not only a projective but an oriented projective reconstruction is obtainable from uncalibrated images.

The advantages of an oriented projective framework are manifold. In addition to the richer set of testable relations between the entities presented in section 2.2, other possibilites gained from the use of oriented projective geometry have been studied and should be mentioned briefly. For instance Werner and Pajdla [2001a] discuss the epipolar geometry constraints in an oriented projective framework. In [Werner and Pajdla, 2001b] this is extended toward matching constraints between multiple images. The necessary modifications to the projective multi-view matching constraints are presented that allow to rule out certain impossible configurations. In [Werner, 2003] it is shown that in the oriented projective framework even five matching points between an image pair, instead of eight, have to obey certain constraints. This ideas are not followed any further here, as the calibration is usually assumed to be given (cf. section 1.3.1).

This section summarized some previous work on projective geometry that is not explicitly focused on uncertain entities. Especially the previous work on oriented projective geometry, which is a major building block of the contribution of this work, and its implications in the computer vision context has been reviewed.

1.2.3 Previous work on data structures and algorithms

The vast increase and cheap availability of computational power in recent years has facilitated many new techniques to become applicable. However, the transition from small toy problems toward the ability to cope with huge real-world datasets is still a major challenge, which has not been very popular within the computer vision research community. One contribution of this work is a search tree data structure, which enables the testing of uncertain relations in an efficient manner (cf. section 2.3). Hence, some previous work on such related search tree data structures will be reviewed in the following.

Another aspect of computational efficiency is the question of complexity. In section 3.4.2 a grouping method based on the partition into cliques of a graph will be presented, which is a problem known to be NP-hard. Therefore some work on this topic will be reviewed here as well.

Search trees

First some literature on the topic of search trees will be reviewed.

The most influential textbook for this work on algorithms and data structures is [Cormen *et al.*, 1990], which gives a broad introduction into the whole field of algorithms including search tree data structures and graph algorithms. Furthermore the classical textbooks [Knuth, 1998a] and [Knuth, 1998b] give a very detailed introduction to the field of search trees.

Many databases use a data structure known as B-trees, which have been introduced by Bayer and McCreight [1972]. It is shown, how insertion, deletion and search of elements can be performed with logarithmic complexity. The insertion and deletion strategies of the data structure presented in section 2.3 are adopted from this, so that the depth of the tree is guaranteed to be logarithmic here as well.

Also based on the B-tree Guttman [1984] proposed the R-Tree data structure, which, using bounding boxes, allows to efficiently insert, delete and search for objects with a spatial extend. In [Beckmann *et al.*, 1990] some more sophisticated tree management strategies are proposed, which further decrease the expected running times. The ideas from this data structures, namely using a necessary condition for each descendant, inspired the data structure presented in section 2.3, though more sophisticated necessary conditions than bounding boxes are used to enable any bi-linear statistical hypothesis tests as queries. In [Beder, 2004a] this data structure, presented in section 2.3, together with the representation of line segments in 2d and 3d, presented in section 2.1, and their tests, presented in section 2.2, have been first published.

An alternative search tree for spatial data has been proposed by Finkel and Bentley [1974], who introduced the so-called quad-tree. This data structure is based on subdividing the whole space into smaller portions and putting an element of a certain spatial extend into any such overlapping portion. This is in contrast to dynamically adopting the bounding

boxes like in [Guttman, 1984], so that it is less suited for generalized necessary conditions. Therefore this path has not been followed here.

The query performance of spatial search trees mainly depends on the statistical properties of the data. There exists for instance queries that have the whole database as a result set, so that no index structure is able to increase the query performance. The empirical evaluation of the data structure presented in section 2.3.4 shows only a linear rather than a logarithmic performance increase caused by the selectivity of the queries. A case study on the statistics of "typical"line segment datasets is carried out by Proietti and Faloutsos [1998], who study the selectivity of queries for real world line segment datasets. It is conjectured that line orientation angles are distributed uniformly, while line segment lengths follow a Poisson distribution, which has to be taken into account, when indexing large line segment datasets. The conjecture is verified on large GIS datasets of road and river networks. No such knowledge has been exploited for optimizing the data structure presented in section 2.3 so far, though.

Complexity

Now some literature on complexity theory and specifically the clique partition problem, which occurs in section 3.4.2 and is known to be NP-hard will be reviewed.

The underlying theory of NP-completeness is developed in the classical textbook [Garey, 1979] for decision problems, which introduced a well-known and often cited list of NP-complete problems, where the specific problem of clique partition occurs as GT15 on page 193.

Paz and Moran [1981] extended the theory of NP-completeness toward optimization problems and introduced the notion of non-deterministic polynomial optimization problems. Those problems are not decision problems like the class of the NP-complete, but share the same discouraging runtime properties. Amongst others, the clique cover problem is shown to belong to this class and that it is equivalent to the colorability problem on the complement graph. Hence, some approximation has to be applied to solve the clique partition problem occurring in section 3.4.2.

The textbook [Ausiello *et al.*, 1999] describes various paradigms and techniques to cope with NP-completeness and gives an introduction to the field of approximation algorithms. Some of the techniques presented there have been applied in section 3.4.2 to find a good solution for the occurring clique partition problem.

The ad-hoc methods proposed in [Ausiello *et al.*, 1999] could be expected to be outperformed by approximation schemes specifically tailored for the clique partition problem. Khuller [1997] gives a broad overview on the state of the art of polynomial time approximations for finding minimum weight spanning subgraphs of a desired connectivity. Many special cases are discussed and results mainly for small connectivities are presented. [Khuller and Raghavachari, 1996] tackle the problem of finding spanning subgraphs with a given connectivity requirement. It is pointed out that the problem is NP-hard for any connectivity greater than one, but a polynomial-time approximation algorithm is presented.

Also the work [Goldschmidt *et al.*, 1996] is about the covering of a graph with cliques of a given size. It is shown that this problem is NP-hard for all clique sizes greater than three. Approximation algorithms for the clique size of three and four are given, as well as an approximation algorithm for arbitrary clique sizes. However, the heuristics applied in section 3.4.2 seem to be well suited for the specific graphs encountered, so that this rather theoretical

results have not been followed up here.

This section reviewed some previous work on search trees relevant to the data structure presented in section 2.3 as well as some previous work on the complexity of the clique partition problem. Besides the literature on the complexity of the clique partition problem itself also some work on the approximation algorithms for this specific problem has been reviewed. However, none of the proposed methods are applicable to the practical relevant cases occurring in section 3.4.2, where the ad-hoc heuristics turned out to be most successful.

1.2.4 Previous work on grouping

Grouping has been a key issue in computer vision from the very beginning and is also the central topic of this work. The classical book [Marr, 1982], which is often considered to spawn the whole field of computer vision itself, propagates an image processing pipeline that is able to interpret a scene by stepwise grouping and aggregation of information. Today perceptual organization is a subfield of computer vision by itself and some previous work will be reviewed in the following. However, grouping, as understood in this work, goes beyond the pure aggregation of image features and includes the whole field of identifying significant structure in observed data, which is now known by the key words data mining, unsupervised learning or clustering. Also some work in this area will be reviewed in the following.

Perceptual organization

As mentioned, an important subpart of grouping in computer vision is known as perceptual organization, which is specifically about grouping together low level image features that is conjectured to take place in the human visual system. The classical textbook in this domain is [Lowe, 1985]. Also the textbook [McCafferty, 1990] is about perceptual grouping developed after the human visual system.

A more recent overview on the current state of the art in perceptual grouping is given by Boyer and Sarkar [1999]. The work in the field is classified according to the level of detail considered and the dimension the algorithms operate on. It is pointed out that most previous work on grouping up to now is in the 2d image domain. The grouping tasks presented in section 3.3 include some 2d reasoning, but the focus of this work is grouping in 3d targeted on the application of 3d building reconstruction.

Some frameworks for the task of perceptual grouping have been presented in the literature. Those frameworks can be classified as voting based grouping methods, energy minimization grouping methods and hypothesize and verify grouping methods. In the following the some previous work classified according to those categories will be presented.

Voting based methods: A classical voting based grouping method is known as Hough-transform. Illingworth and Kittler [1988] review the state of the art in grouping based on this classical technique, where for each observation all possible group models in a discrete parameter space are computed. Those group models, which are supported by most observations, are then selected. Hough-transform suffers from the problem that the whole parameter space has to be discretized, which becomes intractable, if the dimension of this space is large. Therefore it is no longer used widely today, as better grouping techniques are available.

One of those better techniques, which is still voting based, is the prominent tensor voting framework, which is presented in detail in the textbook by Medioni *et al.* [2000].

The first publication of this framework is [Guy and Medioni, 1996], which describes a method for the inference of perceptual contours from local features. For each edgel an extension field is considered, where it votes on the existence of a contour of a certain direction. Each pixel therefore collects direction votes from different edgels and a consistency measure based on the structure tensor of the voting field is applied to decide, where dominant directions are present. Those positions together with the dominant direction are then also considered as edgels, so that the contours are closed.

In [Guy and Medioni, 1997; Medioni *et al.*, 2000] this tensor voting concept is extended to the 3d case, where in addition to linear features also plane-like features are detectable. Schuster [2004] showed the applicability of this approach to LIDAR data. In [Tang and Medioni, 2002] the approach is extended toward curvature estimation by using the inferred dominant directions in a second voting phase.

The tensor voting framework is in principle very similar to the grouping procedure presented in section 3.4.4, because it constitutes an agglomerative grouping procedure that aggregates features based on a combined similarity and proximity measure. The approach presented in section 3.4.4 goes beyond the tensor voting framework, though, as it is not restricted to specific geometric entities (such as edgels) and allows the grouping to be based on arbitrary functional group models.

Energy minimization methods: A second alternative grouping framework, which has been described by McCafferty [1990], is based on energy minimization. The idea is to penalize unlikely groupings with a high energy and obtain a minimum of this energy functional. Also Ommer and Buhmann [2003] present a framework for perceptual grouping based on energy minimization. The costs of grouping two entities together for various Gestalt laws are proposed and assuming a hierarchical grouping structure an overall energy function is given. It is proposed to minimize this energy function in a greedy manner, yielding an agglomerative grouping algorithm.

Another method, which is very similar to the approach presented in [Ommer and Buhmann, 2003] and also based on pairwise potentials, is given by Crevier [1999], who describes an algorithm for grouping co-linear line segments. First a connectivity matrix based on a pairwise co-linearity criterion is computed. For each connected component of the graph induced by this adjacency matrix a probability of being in fact a larger segment is computed and used to decide, if it should be kept as group. The optimization differs from approach of [Ommer and Buhmann, 2003], as those groups of segments are then re-inserted into the procedure as single segments and the process is repeated until all segments are found.

Because the energy of such a system is directly related to an encoding of the data, those energy minimization techniques can be seen as minimum description length algorithms. The algorithm presented in section 3.4.4 minimizes the entropy of the system, which is directly related to the description length, so that the same goal is achieved. However, the minimization is controlled in order to prevent the optimization to get trapped in local minima, as could be expected for the greedy optimization proposed in [Ommer and Buhmann, 2003]. This controlled minimization effectively enforces a notion of proximity while the entropy minimization enforces similarity, which shows the connection between the computational framework of Medioni *et al.* [2000] and the energy minimization framework of McCafferty [1990].

Hypothesize and verify: A third common technique often encountered in perceptual organization is the so-called hypothesize and verify paradigm, which are based on generating

grouping hypothesis and verifying them according to some measure. The first aspect of this technique, namely the generation of good hypothesis, is conjectured to be best solved by perceptual grouping by Kim and Nevatia [1999], though the authors do not give any details. Instead the general need for grouping evidence in order to generate higher level hypotheses in the context of the hypothesize and verify paradigm is formulated. Several methods for combining evidence, including neural networks and Bayesian networks, are discussed. It is pointed out that those more rigorous methods are superior to the ad hoc methods usually used for building reconstruction. This is in complete accordance with the results obtained in chapter 4.

How hypothesis generation can be performed is addressed for instance by Clarke *et al.* [1996], who present an approach for finding line segments in images using RANSAC. The focus of the paper is on the efficiency required for real-time applications. Therefore a small region of the image is extracted and divided into a regular grid. On this grid edgels are extracted and a RANSAC procedure is applied to find larger groups among this edgels. Also the prediction of lines from two oriented views into a third for the task of tracking is sketched briefly. This approach is very similar to the RANSAC grouping method presented in section 3.4.3. The problem of enforcing proximity, which has been encountered explicitly in the tensor voting approach and implicitly in the energy minimization approach, is solve in [Clarke *et al.*, 1996] through the use of a small window in the beginning, though the choice is motivated by the performance requirement there.

The RANSAC approach for grouping has also been used in [Beder and Förstner, 2006a; 2006b] for the task of finding cylinders in point clouds. The problem of proximity has not been addressed there, although it had been one of the major challenges.

Estrada and Jepson [2004] go even further in hypothesis generation and describe a grouping procedure for image line segments based on completely searching the space of all possible contours. Starting from a single line segment, more segments are added based on an affinity score. If the end of a contour is reached, the algorithm backtracks, so that in principle a heuristic search on all possible contours is performed. It is stated that the branching factor is controllable low on real images for the proposed affinity measures. This path of backtracking and complete searching is not followed here.

The second aspect of the hypothesize and verify paradigm, namely the rigorous verification of hypothesis, in the grouping context is the topic of [Heuel and Förstner, 2001]. There the statistical hypothesis test presented in [Förstner *et al.*, 2000], which constitute also the basis for the tests presented in section 2.2, are used as a similarity measure for a reconstruction method based on the grouping of uncertain 3d lines using a hypothesize and verify procedure.

The usefulness of statistical hypothesis testing for the task of grouping has already been demonstrated by Utcke [1998], who used rigorous error propagation and statistical hypothesis testing for grouping 2d edgels into straight lines in images.

It is pointed out in [Heuel and Förstner, 2001] that for the grouping procedure to work properly some notion of proximity between the line segments needs to be applied, which is not explicitly given in the paper. The central importance of this issue for the task of grouping has already been discussed above in the context of the tensor voting framework. One possible rigorous solution for the proximity problem in the framework of statistical hypothesis testing is the entropy based grouping method presented in section 3.4.4, which uses an information theoretic similarity as well as an information theoretic proximity measure derived from the statistical hypothesis tests for evaluation.

Another idea for solving this problem is the use of uncertain oriented 3d line segments, presented in section 2.1, instead of uncertain 3d lines. This idea has been proposed by by

Zhang and Faugeras [1994], although no statistical but geometric distances are used there. The grouping method used in [Zhang and Faugeras, 1994] is very similar to the graph theoretic grouping method presented in section 3.4.2, as clusters of line segments are identified by searching cliques in the graph defined by pairwise distances. This has also been used in [Beder, 2004b], where uncertain line segments were matched across multiple oriented images. In contrast to the work of Zhang and Faugeras [1994], rigorous statistical hypothesis testing was used there.

The previous three frameworks focus on solving some given perceptual grouping problem itself. On a more abstract level Mohan and Nevatia [1992] present a whole perceptual grouping system. Grouping is viewed as a hierarchical process that starting from low-level-features aggregates structure based on certain criteria into higher level structures, which, again, can be grouped on this higher level. Certain criteria for some levels of aggregation derived from the laws of Gestalt psychology are presented. It is pointed out that those image-based criteria should be invariant to the change of viewpoint and exploit some scene structure. This high level view on the grouping problem is not adopted here, though it would of course be useful to integrate the presented grouping steps into an overall system, too.

Other authors have presented hierarchical grouping schemes, too. For instance Venkateswar and Chellappa [1995] proposed a hierarchical matching scheme for finding feature correspondences between two images. In a first step a hierarchical feature graph consisting of lines, vertices and surfaces in the image is constructed. Beginning at the highest level those graphs are matched between the images using various topological, geometric and perceptual criteria for guiding the search process.

Also Lang and Förstner [1996] proposed a hierarchical grouping framework for reconstructing polyhedral objects from multiple images by aggregating features. It is pointed out that a transition into object space at an early stage of processing is crucial, because much more information is available there. This idea is adopted in chapter 4, were the transition into 3d space is performed directly after the feature extraction and all reasoning is performed in the 3d domain. The method presented in [Lang and Förstner, 1996] works by matching the feature adjacency graph between the images using the relational matching techniques proposed in [Vosselman, 1992] and then aggregating the resulting 3d corners into higher level structures by using specific scene constraints. This is in contrast to the method used in chapter 4, where the matching itself is performed using the grouping approach. The subsequent processing of 3d corners into higher level structures is also not the topic of this work.

While the previous work presented in this section so far is focused on the presentation of grouping paradigms, some authors focus on specific geometric entities instead. The most prominent of this geometric entities in the field of perceptual grouping are image line segments, which also have been discussed by many works reviewed above and do not require any special attention here. However, other target models have been used, too, and will be mentioned briefly in the following.

For instance Zisserman *et al.* [1995] discuss some criteria for grouping pixels in images based on the depicted object class. The three classes considered are surfaces of revolution, pipes and polyhedra, each yielding different internal constraints useful for the task of grouping.

Van Gool *et al.* [1998] discuss the usefulness of planar homologies for geometry based

grouping. Two examples for the usefulness are given: shadows of planar structures are related to this structure by a planar homology as are all extruded surfaces, which occur very often in man-made environments.

Also Schaffalitzky and Zisserman [1998; 2000] are concerned with planar structures. They present an approach for grouping repetitive structures in images, which occur on planar surfaces. The image-to-image homography resulting from a translation on the world plane is derived. It is shown how it depends on the two parameters for the vanishing line, one parameter for the translation direction and one parameter for the translation magnitude. A RANSAC based estimation of this four parameters is used to identify the repetitive structures in the image.

In [Kosecka and Zhang, 2005] a method for finding rectangular structures in an image is proposed. Edgels are grouped together yielding dominant vanishing points, which are then used to identify rectangles in the image. A method for recovering the pose of the camera with respect to the world plane defined by the rectangle is given, too.

This section reviewed some previous work on perceptual grouping. Three different paradigms have been identified and examples for those paradigms were presented. Then some work on grouping systems has been presented and a hierarchical grouping approach was identified as very useful for building a grouping based image analysis system. Finally some previous work focusing on specific object classes for grouping has been presented.

Clustering

While the perceptual grouping approaches, presented in the previous section, all focused on image primitives and sometimes on geometric space primitives, grouping can be seen in the much broader context of data analysis, where it is a key task, to identify observations that somehow belong together. The classical textbook in the field of pattern recognition is [Duda *et al.*, 2001], which gives a very broad overview into the field. Of special interest in the grouping context (as defined in section 3.2) are the so-called unsupervised or clustering methods. Those methods require no training data and divide the data into subsets having common properties, which is basically the same task defined as grouping in section 3.2. The major difference to the grouping task presented here must be seen in the existence of a specified target group model, so that grouping can be interpreted as a sub-problem of clustering. An additional difference of clustering to the perceptual grouping tasks presented in the previous section can be seen in the fact that clustering is not limited to geometric observations. However, the grouping problem defined in section 3.2 is in principle not restricted in this sense, either. Two other textbooks in the domain of pattern recognition that influenced this work are [Ripley, 1996] and [Mitchell, 1997].

The importance of minimum description length has already been stressed in section 1.2.1. As already mentioned there, this technique is of vital importance for many data analysis tasks, especially in the absence of other information. For instance Kontkanen *et al.* [2004] present a MDL framework for data clustering. Grouping objects is achieved by using as much group structure as possible, which is expressed by the maximal possible compression of the data. The normalized maximum likelihood code estimated from the data is used for compression, yielding a natural choice of the model complexity term for a given model parameterization.

This approach however, like every pure clustering approach, does not exploit any target group model, which is the major difference between clustering and the grouping problem presented here.

In contrast to this, knowledge of target groups is used in [Beder, 2005], where a framework for grouping observations is proposed, which balances the entropy minimization approach of MDL with the maximum entropy inference paradigm. By analyzing the Kalman filter update equations, the entropy change incurred by including a new observation into a group is quantified. An analogy to statistical hypothesis testing is established and bounds on the admissible entropy updates are derived. The ideas presented in [Beder, 2005] are very closely related to the information theoretic grouping approach presented in section 3.4.4. The major difference is the use of an explicit functional model instead of the implicit functional model presented in section 3.4.4.

Also Kemp and Drummond [2005] exploit a functional target model by proposing a method for grouping observations based on a rank analysis of the normal equation matrix. It is stated that usually the normal equation matrix is sparse, i.e. certain groups of parameters only depend on certain groups of observations. By partitioning the normal equation matrix into those groups, the efficiency is increased, because the subgroups can be treated independent from each other. This again shows the close connection of grouping to the redundancy numbers, which also can be observed in section 3.4.4.

Other clustering techniques are purely based on the pairwise distances between the observations. For instance Hofmann and Buhmann [1997] presented a method for clustering data into groups solely based on their pairwise distances. The approach uses an expectation minimization scheme like the grouping scheme presented in section 3.4.1 and employs the technique of deterministic annealing, which is a maximum entropy method already discussed in section 1.2.1. The drawback of the procedure is, like the expectation minimization scheme presented in section 3.4.1 that the number of groups must be known in advance.

Also based on pairwise distances Amir and Lindenbaum [1998] presented a generic graph theoretic grouping framework. They model the objects as vertices of a graph and the similarity (with respect to the grouping task) as edges. The optimal grouping is then defined as a minimal clique partition of that graph. Furthermore a technique for enhancing the quality of the edge weights is proposed to enhance the possibilities of the applied heuristic search procedure. This is very similar to the grouping method presented in section 3.4.2. The difference to this approach is the fact that the vertices of the graph correspond to the actual observations in [Amir and Lindenbaum, 1998] and not to the candidate group models. If the observations and the candidate group models coincide, which is the case for some grouping tasks with the consistent, sufficient and irreducible subsets of observations (cf. section 3.2) having size one, the two approaches are equivalent.

Also graph based and based on pairwise dissimilarities is the normalized-cut algorithm proposed in [Shi and Malik, 2000]. The problem of image segmentation is formulated as finding groups in a weighted graph, so that the similarity within the groups as well as the dissimilarity between the groups is maximized. It turns out that finding this graph-cuts can be efficiently computed by solving a generalized eigenvalue problem on a normalized adjacency matrix of the graph. This idea is not exploited any further here.

This section reviewed some previous work on clustering methods going beyond the pure

perceptual grouping task. It has been pointed out that grouping can be seen as sub-problem of clustering and some minimum description length methods as well as some methods based solely on pairwise distances have been presented.

1.3 Previous work on required pre-processing

In chapter 4 the presented representations of uncertain oriented projective entities and the presented grouping techniques will be shown to be applicable in the domain automatic scene reconstruction from sets of images. The cameras are in this application assumed to be calibrated and oriented. Furthermore, image features are assumed to be extracted and given together with their uncertainties. In the following some techniques should be reviewed that are necessary to obtain the required input data.

1.3.1 Previous work on calibration

As already mentioned in section 1.2.2, it is necessary to work with images from calibrated cameras in an uncertain oriented projective framework. Although Hartley [1998] has shown that an orientation preserving reconstruction is in principle possible without knowing the calibration of the cameras by just enforcing all scene points to be in front of all cameras, the calibration is required due to the representation of the uncertainty. The uncertainty of the entities is represented here by the second moments in oriented projective space (cf. section 2.1). Those second moments are only able to reasonably reflect the probability distributions of the uncertain entities, if the perspective distortions are low. This is only achievable, if the calibration of the cameras is known. Hence, some calibration techniques that could be used to obtain reasonable input data should be reviewed in the following.

While in classical photogrammetry the camera calibration has been acquired using physical measurements of the cameras itself in a laboratory environment, the calibration techniques favored in computer vision that are of course nowadays also used in photogrammetry fall into two categories: the first group of techniques use images of a known calibration object or a known camera movement while the second group of techniques use weak assumptions on the internal camera parameters itself to obtain the full set of internal camera parameters.

Starting with the first group of techniques, the most prominent paper in this area is by Zhang [1999], who proposed a very simple and easily applicable method. By taking several images of a planar checkerboard pattern with at least two different orientations it is shown, how all internal camera parameters including radial distortion can be estimated. The big advantage of this approach is the fact that it is implemented in Intel's open computer vision library and there also exists a MATLAB toolbox, which are both very common in the computer vision community. Furthermore a planar checkerboard pattern is easily produced, so that no specific laboratory environment is required to obtain a good calibration of the camera.

The second group of calibration methods, also known as self-calibration, use only weak constraints on the intrinsic parameters of the camera. The most prominent approach has been proposed by Pollefeys *et al.* [1996b; 1996a; 1997a], who have shown, how the so-called modulus constraint can be used for affine self-calibration in the case that the internal camera

parameters stay constant over the image sequence. The advantage of using the modulus constraint is that the non-linear optimization required is only over a three-dimensional parameter space. In [Pollefeys and Gool, 1997b; 1997c; 1999] this idea is even extended, by no longer requiring a fixed first reference image. Thereby the modulus constraint can be used for each image pair reducing the minimal number of images required from four to three and increasing the robustness of the procedure. In [Pollefeys *et al.*, 1997b; Pollefeys and Gool, 1997a; Pollefeys *et al.*, 1998a; 1999c] several metric self-calibration techniques relying solely on the assumption that the skew is zero over the whole image sequence are reviewed. It is concluded that it is indeed possible to obtain a metric reconstruction from an image sequence under this weak assumption. In [Pollefeys and Gool, 2000a; 2000b] some insight into critical motion sequences for self-calibration are given. A whole system for metric reconstruction based on self-calibration from an uncalibrated image sequence is presented in [Pollefeys *et al.*, 1998b; 1998e; 1998c; 1998d; 1999a; 1999b; 2000b; 2000a; 2000d; 2000c; Pollefeys and Gool, 2002; Pollefeys *et al.*, 2002a; 2003]. A very comprehensive report on this system can be found in [Pollefeys *et al.*, 2004].

The applicability of this self-calibration techniques is somehow limited and an awareness of the limitations is strongly required to apply them. Besides the need for image sequences and non-changing cameras (including focal length), the problem of degeneracies resulting from planar objects are studied in [Pollefeys *et al.*, 2002b]. As a solution, it is proposed to partition the image sequence into sub-sequences seeing planar structures and sub-sequences seeing 3d structures as described by [Torr, 1997] and applying different self-calibration methods for each case.

In [Repko and Pollefeys, 2005] the problem of processing long image sequences for self-calibration and orientation is discussed. Using the geometric robust information criterion (cf. Torr, [2002]) triplets of key-frames are selected from the sequence and local reconstructions are used to overcome the problem of projective drift, which can significantly decrease the performance of self-calibration.

In this section two major contributions on calibration, namely using a simple calibration object and the so-called auto-calibration, were reviewed due to the crucial importance of calibrated cameras for the uncertain oriented framework subject of this work. Some work on the limitations of auto-calibration has been presented as well, showing the limitations of this approach. It must be strongly recommended to use the first approach with the calibration pattern whenever possible, because it is often then simplest and most robust choice. Having now discussed methods for obtaining the calibration of the camera, the next section will be about obtaining the orientation of the cameras, which is the second prerequisite of the reconstruction application presented in chapter 4.

1.3.2 Previous work on structure from motion

The recovery of 3d structure from corresponding points in multiple images has always been a key issue in computer vision and photogrammetry (e.g. [Horn, 1986], [Faugeras, 1993] or [Hartley and Zisserman, 2000]) and is also the goal of the polyhedral object reconstruction application in chapter 4 of this work. Given corresponding image features, it is possible to recover the scene structure as well as the exterior orientations of the cameras. The geometry related sub-problems of this task are summarized here under the keyword structure-from-motion. In the following some prior work on this subject will be reviewed for two reasons:

first, those problems occur also in some of the grouping applications presented in section 3.3 and second, the orientations of the images are required as input data of the building reconstruction application presented in chapter 4.

First some previous work on pure relative orientation of two images will be presented followed by a short section on pure triangulation. Those two sub-tasks are often solved simultaneously, which is summarized under the topic bundle-adjustment that will be presented in the subsequent section. This section will be concluded by a review of prior work on the topic of simultaneous localization and mapping, which is concerned with the specific demands of bundle-adjustment for image sequences.

Relative Orientation

The first crucial problem encountered in structure from motion is the determination of relative orientation between an image pair for given point correspondences. Like calibration, presented in the previous section, this is an important pre-processing step that must be performed before the grouping techniques presented here are applicable. All approaches presented here assume the calibration to be known.

Horn [1990] described an iterative approach for finding the relative orientation of an image pair. The problem of finding an appropriate error metric for the co-planarity condition is discussed. It is conjectured that the proposed method does not rely on good initial values. The critical surfaces of the method are analyzed as well. As such iterative procedures always suffer from possible inappropriate choices of initial values, non-iterative procedures have been developed. For instance Philip [1996] presents such a method for determining the relative orientation of two calibrated cameras from six points as well as a method for five points. Both methods first compute the basis of the space of essential matrices, where the solution can be found. Using additional cubic constraints, a linear equation system is derived for the three-dimensional space derived from six correspondences. In case of five correspondences the cubic constraints are used to derive a 13th-degree polynomial, whose roots give solution candidates for the essential matrix. In [Philip, 1998] the critical point configurations for the algorithms presented are studied. Amongst other results it is shown that all those algorithms fail if the points are all co-planar. Solving this degeneracy, Nistér [2003; 2004] gives an efficient solution for determining the relative orientation of two calibrated cameras from the minimal set of five points. From the five point correspondences the basis of a four-dimensional space, where the solution must lie, is calculated. Using additional cubic constraints on the essential matrix, this up to ten solutions are calculated by finding the roots of a 10th-degree polynomial. It is shown that there exist indeed cases, where all ten roots correspond to feasible solutions. In contrast to [Philip, 1996] the algorithm is capable of dealing with planar scenes. In [Stewenius *et al.*, 2006] a different implementation of the algorithm based on the computation of a Gröbner basis is presented. Instead of finding the roots of a 10th-degree polynomial the eigenvectors of a 10×10-matrix are computed to find the up to ten real solutions. This last algorithm must be considered as the method of choice, because its numerical stability is best and it can deal with planar scenes, which often occur in practical applications.

As mentioned above, all those approaches need the calibration of the camera to be known. In case of a zooming camera this is often impractical due to the changing focal length. Tackling this problem Stewenius *et al.* [2005a] proposed an algorithm for computing the relative orientation of an image pair without knowing the focal length from six corresponding image points. The polynomial constraints are formulated and it is shown, how to obtain the up to 15 solutions using Gröbner basis methods. However, this algorithm is again unable to

cope with planar scenes.

One problem not solved by all pairwise relative orientation methods is the computation of a single consistent set of orientations for multiple images, which is ultimately required. A solution is proposed for instance by Goldberger [2005], who presents an algorithm for obtaining a consistent set of projection matrices from pairwise fundamental matrices. Constraints between projection matrices and fundamental matrices are developed and a stepwise procedure for obtaining all orientations is given. This set could be used directly as input data for the grouping methods presented here or alternatively could be refined using the bundle adjustment techniques presented in section 1.3.2.

Some more or less simple methods for obtaining relative orientations of image pairs have been presented. The relevance for this work stems from the fact that orientations are required as input data for the building reconstruction application presented in chapter 4. If the focal length is known, the method presented in [Stewenius *et al.*, 2006] should be used, as it is robust and able to cope best with nearly planar scenes that often occur in practice. Furthermore, the orientations should be refined using the bundle-adjustment techniques that will be reviewed in section 1.3.2.

Triangulation

The problem of triangulating points from given correspondences and given camera orientations occurs as sub-problem when grouping 3d lines into 3d points (cf. section 3.3). It is, however, not trivial and recently some promising new approaches have been presented. The importance for this work stems from the fact that the iterative triangulation method presented in [Heuel, 2004], which has been used here, sometimes suffers from solutions being only local minima of the cost function or even divergence of the optimization resulting in missed matches. Different methods for obtaining initial values could therefore improve the results.

One approach introduced by Stewénius *et al.* [2005b] gives a direct solution for finding the maximum-likelihood estimate of a 3d point from exactly three cameras. Using the polynomial equations resulting from the minimization of the L_2-distance between the reprojected and the observed image points, the up to 47 solutions are computed using a Gröbner basis method showing the problems complexity. As the requirement of exactly three images is too restrictive to be applicable here, this path has not been followed, though.

A different method is proposed by Kahl [2005], who introduced a framework for solving geometric structure and motion problems using the L_∞- instead of the L_2-norm. By stating the problems as quasi-convex optimization problems it is possible to efficiently find a global optimal solution. It is stated that though the L_∞-norm is extremely sensitive to noise, this approach is comparable to classical methods and could also be used to find initial values for an iterative L_2-norm optimization. Due to the complexity and the sensitivity to outliers of this method, it has not been applied here. However, one could think of improving the estimation results of Heuel [2004] for instance by initializing the estimation with the L_∞-norm solution in case of divergence. This path has not been take here either, though.

This section reviewed two promising approaches for improving the estimation results obtained in chapter 4 by improving the initial triangulation. The improvement that could be

gained by this has not been studied here. Apart from the desirability of optimal results, the actual improvement in practical applications can be assumed to be minor, though, because it could be observed that divergence occurred only on a very small fraction of estimations. Hence, no path toward this direction has been followed here.

Bundle adjustment

The previous two section reviewed techniques for obtaining pairwise relative orientations on the one hand and given those orientations scene points on the other hand. It is often favorable to combine those two geometric tasks into one single step, as statistical optimal solutions are only obtainable like this. However, the methods presented above are still applicable for obtaining initial values for the iterative combined methods known as bundle-adjustment. A very broad overview over the current state of the art of bundle-adjustment is given by Triggs *et al.* [2000]. The paper contains a detailed and consolidating discussion on the topics of choosing an appropriate cost functions, optimization techniques, approximations, recursive estimations, gauge invariance and quality assessment.

From the discussion in [Triggs *et al.*, 2000] follows that the generic problem of bundle-adjustment is a difficult problem requiring a lot of effort. Often simpler approaches are possible. Of particular importance among those simplifications is an approach presented by Rother and Carlsson [2002], who propose a linear algorithm for simultaneously estimating scene points and camera positions in a projective frame. Linearity is achieved by transforming the image coordinate systems using a single reference plane that is visible in all images. This is a weak assumption for many applications, which could also be fulfilled by the plane at infinity if the mutual rotations of all cameras are known (for instance from the relative orientation techniques presented in section 1.3.2).

It is widely recognized in the computer vision community that a final step of every structure-from-motion algorithm should be a bundle adjustment, in order to obtain optimal results. The grouping techniques presented in this work require image orientations to be known in the first place, so that here a bundle adjustment is also required as a pre-processing step. As the accuracy demands might not be that high in the pre-processing, either the relative orientations obtained using the techniques of section 1.3.2 are used directly or alternatively the simple approach presented in [Rother and Carlsson, 2002] should be used.

The bundle-adjustment techniques presented so far are only concerned with point features. As lines and line segments are important image features and will be used extensively in this work, too, some previous work on bundle-adjustment techniques based on lines and line-segments will be reviewed as well.

A broad overview of the state of the art of structure and motion recovery from lines can be found in [Bartoli and Sturm, 2005; 2003]. In there mainly the topic of 3d line representation and its implications on various reconstruction algorithms are discussed.

Early work on the topic of bundle adjustment using line segments has been done by Taylor and Kriegman [1995]. The re-projection error of the scene lines is measured along the extracted image line segments in each image. The optimum of this cost function depending on the space line and camera positions is found via gradient descend like in the classical point-based bundle-adjustment.

Also Zhang [1995] presented a method for computing structure and motion from corresponding line segments. In contrast to [Taylor and Kriegman, 1995] the overlap between

matched line segments is measured as cost function. It is pointed out that, as line segments rather than lines are considered, it is possible to solve this problem with only two perspective images.

An approach for simultaneous bundle adjustment from point and line features based on the relations presented in section 2.2 as cost function is presented in [Luxen and Förstner, 2001], where an algorithm is proposed that optimally estimates the internal and external orientation of the camera from observed scene points and scene lines of known position in the image as well as in the object space.

Also Bazin and Vezien [2005] presented a structure-from-motion algorithm that integrates geometric entities such as points, line segments and also rectangles, their uncertainties and mutual relations and a parameterized camera motion model. The focus of the paper is on elimination redundant parameters and on estimating the reduced model parameters optimally yielding robust reconstruction and orientation results.

Finally also the approach of Rother and Carlsson [2002], which has been favored above, is extended toward lines and planes in [Rother, 2003].

In this section some previous work on bundle adjustment has been summarized. The first part has been concerned with work on bundle-adjustment for point features, while the second part presented some previous work on bundle-adjustment using lines and line segments as features that both occur in the building reconstruction application presented in chapter 4. The relevance of those techniques for this work is two-fold: first, oriented images are required for the grouping results obtained in chapter 4 and second, the results obtained with any structure-from-motion algorithm should be post-processed using bundle-adjustment techniques. However, this latter has not been done presently, as the accuracy of the camera positions in the building reconstruction application is assumed to be very precise for the given aerial images.

Simulataneous localization and mapping

Systems that simultaneously acquire scene structure and camera orientations in an on-line manner from image sequences have become popular recently under the keyword simultaneous localization and mapping. Because this systems present an interesting method for obtaining oriented images and because they also can be considered as competing technique to the reconstruction application presented in chapter 4, some previous work in this area will be presented.

One major advantage of such active systems, which originate in the robotics community, is that they are often free to navigate the environment. For instance Davison and Murray [2002] present a method for simultaneous localization and mapping based on active vision. Using a Kalman filter it is predicted, where features are likely to occur in the scene. The active vision system is then targeted on this features in order to obtain the precise location. Interestingly, the grouping method presented in section 3.4.4 is also based on analyzing the Kalman filter, but prefers to choose more uninformative observations to be more robust to outliers. The choice in [Davison and Murray, 2002] on the other hand is the most informative feature, which could distort the result significantly, in case it is a leverage point.

This active vision system is not restricted to be used with active hardware as has been demonstrated for instance in [Davison, 2003], where a real-time localization of a mobile single camera is presented. Using a simple motion model for the camera the location of interesting

scene features is predicted using a Kalman filter. Simultaneously interesting and stable scene features are mapped to be used for later orientations. This system has a high potential to be used to obtain sets of oriented images in a fully automatic and efficient manner.

Other groups have also proposed approaches for simultaneous localization and mapping. For instance Se *et al.* [2005] presented an automatic localization method using SIFT features (cf. Lowe, [2004]). A map is generated, which contains for every scene point a SIFT descriptor. Localization is performed by matching the image features against the map features. Two approaches based on RANSAC and on Hough transform are compared.

There exist some drawbacks, though. One major disadvantage of the current techniques are the huge memory requirements that limit the applicability to small problems. Sim *et al.* [2005] report on a purely stereo vision-based system for simultaneous localization and mapping. Landmarks are represented using SIFT features as described in [Se *et al.*, 2005] and the motion is estimated using a particle filter. It is reported that running times for large environments are still to high for the applicability of systems that do not take advantage of non-vision sensors. Hence, classical wide-basline systems combined with other sensors are still the best choice for the task automatic orientation and should be used whenever the problem size is too large.

One central issue in simultaneous localization and mapping systems is the proper selection of key-frames in order to obtain stable reconstructions. The problem is not only the computational and memory efficiency, but also the issue of stability resulting from the very small base-lines in image sequences.

To cope with this problem, Schaffalitzky and Zisserman [2002] presented a collection of algorithms and heuristics to efficiently find spatial clusters in unorganized image sets that enable the computation of orientations for large databases. Features and invariant descriptors are detected and from this putative correspondences between image pairs are computed. Based on this and the robust estimation of the epipolar geometry the matches between the image pairs are refined and connected components of image sets and feature tracks are identified. Finally auto-calibration and bundle-adjustment are employed on the connected components to recover the interior and exterior orientation of the whole image set.

Also Koch *et al.* [1999b] presented a method for robust calibration and orientation as well as surface reconstruction from large image sets. By analyzing the estimated epipoles and residuals of estimated homographies of image pairs the topology of the camera positions is recovered. Using this topology information, suitable subsets of images are selected for calibration, orientation and reconstruction. This pre-processing is of vital importance for coping with large image sets.

Another idea for key-frame selection is presented by Thormählen *et al.* [2004], who proposed to use the trace of the expected covariance matrix of the bundle adjustment to select those frames yielding the highest accuracy in the estimation. This is very similar to the approach presented in [Beder and Steffen, 2006], where a method for fixing the scale of a 3d reconstruction is presented. The roundness of the expected covariance matrices, which is related to the condition of the normal equation system for reconstruction, is maximized, in order to obtain frames that enable a stable reconstruction.

Depending on the application the techniques presented in [Schaffalitzky and Zisserman, 2002] and [Koch *et al.*, 1999b] or the techniques presented in [Thormählen *et al.*, 2004] and [Beder and Steffen, 2006] might be more applicable. In any case, the proper selection of key-frames is of vital importance for using image sequences as a mean of generating sets of

oriented images that are required for the reconstruction application of the grouping methods presented in chapter 4.

In the previous sections a brief overview on the geometric aspects of recovering scene structure as well as camera positions from sets of images, which is summarized under the topic of structure-from-motion, has been presented. The need for oriented images motivated most of the presentation including the sections on relative orientation, bundle-adjustment and SLAM. The section on triangulation and part of the section on bundle-adjustment were motivated by the fact that it occurs as sub-problem in some of the grouping tasks presented in section 3.3. Now the geometric prerequisites for the reconstruction application presented in chapter 4 have been discussed. The following section will be on the image processing requirements, namely the extraction of suitable features.

1.3.3 Previous work on feature extraction

The final crucial pre-processing step for the proposed application of the grouping algorithms is the extraction of features from the images. As the grouping results are critically dependent on good feature extraction, some previous work in this area will be reviewed in the following. First feature extraction methods focusing on image points followed by methods focusing on image lines and finally on image regions will be presented. As a major influence factor on the results of feature extraction the scale of the features and the images has been identified, hence some previous work focusing on this topic will be presented as well.

Points

The most informative image features for the task of bundle-adjustment are image points, as has been seen in the previous section. Therefore a large amount of work has been done in this area. An early overview on the topic of interest point detection has been given by Deriche and Giraudon [1993].

A more recent overview on the current state of the art of interest point detection is given by Schmid *et al.* [2000] and even more recently by Mikolajczyk *et al.* [2005]. In [Schmid *et al.*, 2000] the two quality criteria repeatability and distinctiveness under varying image transformations are proposed to evaluate the feature detectors. In [Mikolajczyk *et al.*, 2005] also the correctness of point matching between images is used as a quality criterion. Of those quality criteria only the repeatability is of relevance for the geometric grouping techniques presented here, because the feature matching demonstrated in chapter 4 does not use any radiometric information from the images at all and hence cannot benefit from radiometric distinctiveness.

Other quality criteria have been studied as well. For instance the performance of different interest point detectors is compared on the basis of database retrieval results by Mikolajczyk and Schmid [2003; 2005], as has already been proposed as quality criterion by Schmid and Mohr [1997; 2000]. It is found in this very popular work that the influence of the specific detector is negligible and that the descriptor of Lowe [1999; 2004] outperformed the other descriptors. In [Moreels and Perona, 2005] the same analysis is done explicitly for objects with significant 3d-structure. It is found that no detector/descriptor-pair performs well under viewpoint changes of more than $25-30^\circ$. This latter result can be helpful in desiging imaging setups, while the fact that the influence of the detector is negligible, must be doubted for the

application presented in chapter 4, which turned out to heavily depend on the quality of the feature extraction.

The most well-known interest point detector is presented in the classical paper by Harris and Stephens [1988], who analyzed the intensity change incurred by a shift, which must vanish in homogeneous regions. The well-known combined corner and edge detector based on the determinant and the trace of the structure tensor is derived. Basically the same idea has been published earlier by Förstner and Gülch [1987], who presented a feature detector based on the quality of least squares matching. By analyzing the expected accuracy, corner points together with their covariance matrices are extracted. The major advantage of this approach is the computation of uncertainties of the image features as a byproduct, which are required in the framework presented here. Hence, the interest point extraction is performed using this technique here.

The expected accuracies of the image features can only be interpreted properly, if the noise level of the image is known. In [Förstner, 1998] a method for estimating this noise level is presented that will be used here as well to extract the required uncertainties from the images.

If the noise level of the image is unknown, the condition of the structure tensor, which is independent of the noise level, can be used as a criterion instead. This has been proposed by Shi and Tomasi [1994], who presented a method for extracting interest points based on the condition of the expected covariance matrix of the estimated transformation between two frames. The explanation given there is the idea that, if the normal equation system of the tracker of Lucas and Kanade [1981] for an affine motion model is well conditioned, then a certain feature is well suited for the task of tracking. However, the absolute accuracies are thresholded there, too, so that the noise level must be known as well. This is in principle the same argument given in [Förstner and Gülch, 1987], hence nearly the same criterion is derived. Also Triggs [2004] presented such an interest point detector, but in a broader framework that generalizes toward generic motion and illumination models. This generalization could be used to obtain more accurate uncertainty estimates for specific applications. However, this path has not been taken here and the method presented in [Förstner and Gülch, 1987] in conjunction with the noise estimation presented in [Förstner, 1998] is used to obtain the results presented in chapter 4.

Lines

The second important features that frequently occurs in man-made environments are image line segments. Because line matching is another important grouping application presented in chapter 4, some prior work on this topic will be reviewed as well.

The most well-known edge detector was proposed by Canny [1986], who derived a detector by optimizing a cost-function that explicitly models the criteria of optimal detectability, optimal localization and suppression of multiple responses. As a post-processing step the so-called hysteresis thresholding ensures that few spurious small line segments occur in the results.

The same goal of suppressing spurious responses in textured image areas is followed by Heitger [1995], who presented an improved edge detector based on linear filters. In addition to using the anisotropic filter responses in different directions the first and second derivatives of the response function along the orthogonal direction are used to enhance or suppress the

original responses. Thereby isolated edges are favored over textured areas.

This problem of noise sensitivity of edge detectors is also tackled by Smith and Brady [1997], who presented the so-called SUSAN edge and corner detector. This detector does not incorporate any image derivatives, but decides on the interestingness of a pixel by measuring the nearby area of equal intensity, which should be one half at edges and less at corners. It is claimed that this measure is very robust, because no derivatives are need, which are known to enhance the noise.

All this three presented edge detectors only indicate certain pixels as belonging to edges. None of them extracts line segments itself nor does any of them extract uncertainties of the results. The first issue of extracting line segments can be solved by chaining edge pixels as presented for instance in [Fuchs, 1998]. The second issue of extracting uncertainties that is vital for the uncertain oriented projective framework of this work can be solved by analyzing the structure tensor as it has been done for the point extraction. This strategy has been proposed by Förstner [1994], who extended the idea of [Förstner and Gülch, 1987] toward segmenting the image content into point-like, line-like and region-like areas using the eigenvalues of the structure tensor. This feature extraction operator is used here, as it is capable of extracting points as well as line segments together with their uncertainty that are required in the grouping framework presented in section 3.2.

Regions

The third type of features that can be extracted from images besides points and lines are image regions, which are connected image areas that are homogeneous according to some criterion. The feature extraction framework presented in [Förstner, 1994] and [Fuchs, 1998], which has been discussed in the context of line extraction in the previous paragraph, basically segments the image into point-like, line-like and region-like areas. The relevance of image regions for the task of matching is the fact that theoretically only regions cover an image area to extract radiometric information from. On the other hand only purely geometric matching algorithms are able to work with image points and image lines. Hence, all matching algorithms, which are not purely geometry based like the one presented in this work, need in fact image regions rather than points and lines as input data, i. e. use at least some image content in the vicinity of the feature points and feature lines. The geometric information contained in regions, however, is limited, so that intensity based approaches are the right choice for processing image regions and geometry based approaches can be best applied for processing image points and image lines. Therefore no region matching algorithm based on the grouping framework will be presented in the following. However, as the competing radiometric matching algorithms only work on (possibly small) image regions, some additional previous work on the extraction of image regions will be presented. Furthermore, region extraction can also be used for point and line extraction by using the image topology, as those features can also be considered as the image content, which is left after all regions have been removed.

One genuine region extraction approach, which has received a lot of attention in recent years, has been presented by Matas *et al.* [2002]. In there a method for extracting maximal stable extremal regions in gray-value images is described. It is conjectured that those regions, where the local binarization is stable over a large range of thresholds, are very well suited for the task of wide baseline matching.

The topology of regions is an important aspect. While the stability criterion of [Matas *et al.*, 2002] uses the image topology alone for extracting good regions, other available information may be incorporated to solve the difficult task of region extraction. For instance

Bretar and Roux [2005] propose to use LIDAR data as additional cue in order to extract homogeneous regions from oriented aerial images. The 3d point cloud is first triangulated and segmented into planar primitives. The planar primitives are then projected into the image and a region merging based on color distances is performed, which takes also the connectivity information from the planar primitives into account. Applications using such additional information, from which also the geometry based reconstruction would benefit, are not the scope here, though.

As mentioned above, regions are often extracted in the vicinity of interest points, in order to facilitate intensity based matching. One such region extractor has been proposed by Tuytelaars and Gool [2004], who describe two methods for finding affine invariant regions around interest points that are suitable for matching. The first method uses the extracted image edges adjacent to the image point to grow a region until a photometric similarity criterion for the covered region is maximal. The second method does not rely on extracted edges to grow this region, but grows the region in circular directions from the interest point. Generalized color moments are used to describe and match the extracted regions between the images.

Also the extraction of interest points in scale-space, which will be discussed in more detail in the following section, can be viewed as region extraction, because feature points in very low-resolution images correspond to larger regions in high-resolution images. The most prominent work in this context, which has received a lot of attention from the computer vision community, is by Lowe [1999; 2004], who proposed to use scale-space maxima as features for the task of matching. Those features, although being points in scale space, may correspond to large circular image regions on the original image resolution. Those circular regions may also be used to define some notion of uncertainty of those features, which would be required in the uncertain framework presented here. However, no theoretical consistent framework for this idea exists so far, so that this path is not followed here.

Scale

As pointed out in the previous paragraph on region extraction, scale plays an important role in feature extraction. The reason for this is two-fold: first, objects usually occur on specific scales in images and second, radiometry based point and line matching algorithms require image information from a neighborhood of the features. The size of this neighborhoods is usually derived from the scale of the features. Although the purely geometry based point and line matching framework presented in section 3.3 and chapter 4 does not require a neighborhood of the features to be known and is therefore able to operate on true 1- and 0-dimensional features, it requires an uncertainty of the features to be known, which is linked to the scale of the feature. Furthermore, the issue of repeatability of features together with the fact that certain features only occur on certain scales require the feature extraction procedure to have some scale-invariance, in order for the presented grouping framework to work. Therefore some previous work on the scale of features is presented in the following.

The importance of scale for feature extraction has been noted early by Koenderink and van Doorn [1987], who studied the local reception of features in the biological visual system. It is conjectured that features are described locally by a set of filter responses to the convolution with higher order derivatives of a Gaussian kernel. Thereby the local scale and geometry is captured in a very simple and localized descriptor for further processing, giving a biological motivation for scale-space features in computer vision.

From a more technical point of view Lindeberg [1998b] discusses the importance of proper

scale selection for the task of feature extraction. It is proposed to use local maxima in scale-space itself for feature detection in order to solve the problem of automatic scale-selection. This approach has received a lot of attention from the computer vision community especially in the context of SIFT features introduced by Lowe [1999; 2004]. The approach has been extended toward edge features in [Lindeberg, 1998a], where an analogous edge detection algorithm is proposed that operates directly in the scale space. Instead of searching edges at a single fixed scale, in this framework edges are defined as one-dimensional curves in the three-dimensional scale-space. In particular the scale is allowed to vary along the edges themselves.

Other criteria for finding features in scale-space have been proposed. For instance Kadir and Brady [2001] argue that the two key aspects of feature detection, namely saliency and scale, are inter-related, and that a good interest point operator should detect salient, i.e. complex, features not at different scales, but detect complexity in scale-space itself. An algorithm is derived that detects features, where the local entropy is maximal in the scale-space. In [Kadir *et al.*, 2004] this approach is extended toward affine invariance by augmenting the single scale parameter with two extra parameters that describe a sampling ellipse instead of a circle in scale-space.

Scale invariant feature extraction does not need to operate on scale-space itself. For instance Mikolajczyk and Schmid [2001] presented a scale invariant interest point detector. First interest points are detected using the approach of Harris and Stephens [1988] on different scales of interest. For each such point the characteristic scale is determined by finding the scale for which the Laplacian at its position is maximal. In [Mikolajczyk and Schmid, 2002] this idea is refined toward affine invariance by first rectifying the image at each interest point using the local structure tensor. As the characteristic scale of each interest point is not needed for the purely geometry based feature matching presented here, this path has not been followed, although an improvement might be expected from detecting features at multiple scales.

In this section previous work on the extraction of features from images has been presented. The first two sub-sections reviewed some literature on the extraction of points and lines that will be needed together with the camera calibration and orientation as input data for the building reconstruction application presented in chapter 4. The competing matching techniques that are based on comparing the radiometric contents of the images, are in principle unable to work with 0- and 1-dimensional features and require either regions or a specific feature scale to be extracted instead. Hence, some previous work on this topics has been presented as well. In the following section those competing matching techniques will be discussed.

1.4 Previous work on competing techniques

In chapter 4 an application of the presented grouping framework for the task of matching point and line features across multiple oriented images will be demonstrated. The approach is solely based on geometric cues in this framework and competes with the feature matching approaches that are based on the radiometric image content. Therefore some previous work on the topic of feature matching will be presented in the following. Furthermore, the application domain of building reconstruction from aerial images, which has also been chosen in chapter

4, has always been of special interest in the photogrammetry domain and some previous work on this topic will be presented as well.

1.4.1 Previous work on feature matching

As pointed out before, the proposed grouping framework is applicable to the task of matching point and line features across multiple views on a purely geometric basis. In contrast to this, much work in the computer vision literature has been concerned with the matching of image features based on the image content (e.g. [Horn, 1986] or [Faugeras, 1993].

Also Torr and Zisserman [1999; 2000] advocate the use of features for structure and motion estimation in contrast to dense matching techniques. Those dense matching techniques are not considered here, but a broad overview on the current state of the art in this field has been given by Scharstein and Szeliski [2002], who compared the performance of various techniques on many test images with known disparity maps.

The field of image-based feature matching can be viewed as competing technique to the geometric grouping approach presented here, as the same problem is solved. Due to the usually much higher redundancy, the radiometry-based methods outperform the purely geometric methods if only few images are used, as will also be seen in chapter 4. However, the two approaches are complementary, so that better results are obtainable by combining both approaches, which must be strongly recommended in practice. The results of the purely geometric matching presented in chapter 4 show the importance of geometric cues, though, as it will be seen that matching is indeed possible without looking at the images at all, if the geometry is known.

In the following some competing (or supplementary) previous feature matching approaches will be reviewed. Following the outline of the feature extraction section, first matching algorithms focusing on image points followed by methods focusing on image lines will be presented. The section will be concluded with some brief remarks on the connection of the grouping framework with prior work on region matching that could be considered as part of the dense matching techniques not considered in this work.

Points

Starting with the point features it has already been noted in the section on feature extraction that those have received a lot of attention in the field of photogrammetry and computer vision mainly due to their suitability for the task of bundle adjustment. In order to be able to apply those techniques, features need to be matched between the images, which must be considered as one of the most challenging problems in computer vision that has not been solved in a satisfactory manner, yet. However, in very constraint imaging situations correlation based techniques described for instance in [Faugeras, 1993] or [Vosselman *et al.*, 2004] have turned out to be quite successful.

Already Helava [1976], who reviewed several image point matching techniques, concluded that multi-level correlation had been superior to all other techniques considered in his review.

A more recent overview on the current state of the art of wide baseline feature matching has been given by Gool *et al.* [2002], who discussed the various sub-tasks and the problems encountered in this area.

In the following several point matching techniques will be discussed. First some work specifically concerned with comparison-based wide baseline matching will be presented, which

is very well suited to be combined with the geometric approach presented in section 3.3. Thereafter some previous work on tracking, i.e. assuming short baselines, will be presented. In contrast to the comparison-based techniques descriptor-based matching techniques have become popular recently and some previous work on this topic will conclude this section.

A purely radiometric wide-baseline point matching algorithm has been presented by Baumberg [2000], who proposed to compute affine invariant windows from the local structure tensor at each feature point and generate correspondences based on the distance between those normalized image patches. This idea is completely complementary to the purely geometric approach presented here and is therefore very well-suited to be combined. However, this has not been exploited, as the goal of chapter 4 is to demonstrate the power of the purely geometric grouping approach and not to devise an optimal algorithm.

The same idea has been used by by Xiao and Shah [2003], who propose to estimate an affine transformation between each pair of interest points following the idea of Lucas and Kanade [1981] and use the residuals of this estimation to generate putative correspondences. The close connection to the approach of Baumberg [2000] results from the close connection between the structure tensor and the technique of least squares estimation that has already been used in [Förstner and Gülch, 1987].

The main advantage of this residual-based image point comparison technique is that it easily allows to model various assumptions on the local geometric and radiometric transformations between the images. For instance Georgescu and Meer [2004] describe a least-squares parameter estimation based method for matching points across two widely separated views that also differ significantly in illumination. The local 2d homographies together with the color transformation between the images are modeled and their parameters are estimated using a robust M-estimator. The drawback of incorporating many parameters in the model is, apart from possible problems with the estimation, that the residuals decrease, worsening the distinguishability between the features. However, the geometric matching techniques presented here could compensate for this weakness.

Using geometry for this compensation has already been proposed by Pritchett and Zisserman [1998b], who presented an approach for wide-baseline matching, which is based on estimating local homographies for each interest point pair, too. Using those initial putative correspondences, a RANSAC procedure is applied for finding dominant planes in the scene. The correspondences not consistent with any dominant plane are removed and again a RANSAC procedure is applied to compute the fundamental matrix, retaining all point matches consistent with the epipolar geometry. In [Pritchett and Zisserman, 1998a] the approach of [Pritchett and Zisserman, 1998b] is extended toward three views. Between all three pairs of images the correspondences are computed according to [Pritchett and Zisserman, 1998b] and the corresponding triples are determined from the resulting matched point pairs by checking, whether they have any point in common. Finally the trifocal tensor is estimated using a RANSAC procedure from those triples, retaining all point triples consistent with the trifocal geometry. In contrast to this approach, the geometric matching techniques presented in section 3.3 are not restricted in the number of views enabling it to exploit more redundancy in the presence of more than three views.

All matching techniques presented in the previous paragraph are based on comparing small image patches between two views. Other comparison methods have been proposed. For

instance Carneiro and Jepson [2002] present a method for matching point-features between two images using normalized phase correlation as similarity measure. Rotational invariance is achieved by integrating over all possible rotations.

Another idea is to use image topology. For instance Tell and Carlsson [2000] proposed an algorithm for matching intensity profiles between two images. Interest points are computed and for random pairs of them the intensity profile on the line between them is extracted. Using a voting scheme those profiles are matched across the images by comparing scale-invariant features based on their Fourier transform. In [Tell and Carlsson, 2002] this topological idea is augmented by penalizing point matches, where the cyclic order of profiles to neighboring interest points differs.

The topology of interest points has also been used by Brown and Lowe [2002], who proposed a method for matching small adjacent groups of point-features between two images. Interest points, which are the extrema of the Laplacian in scale-space, are extracted and for each such interest point the 2, 3 or 4 nearest neighbors in scale-space are used to estimate a similarity, affinity or homography into a canonical reference frame. Finally this resampled reference frames are correlated between the images, which is very similar to the approach presented in [Baumberg, 2000], although the arrangement of interest points itself rather than the structure tensor is used to capture the local geometry.

Thus far algorithms have been discussed, which make no assumption on the length of the baseline between the images and are therefore called wide-baseline methods. If a short baseline can be assumed, then the matching problem is also known as tracking, which has also received a lot of attention in the computer vision community due to the wide availability of image sequences and the possibility to process them in real-time. This is in contrast to the geometric grouping methods presented in section 3.3, which even require large baselines in order to work properly. However, some work in this field will be reviewed, as it can be considered as competing technique if image sequences are available.

The most well-known tracking algorithm has been proposed by Lucas and Kanade [1981], who presented the now classical least-squares matching technique, where an iterative algorithm for computing the translation between two signals was proposed. The squared difference of the two signals is minimized using a linear approximation and an explicit expression for the translation is derived. Starting from an initial estimate of the translation it is then possible to refine the translation vector iteratively. Although this idea has occurred earlier in in [Helava, 1976], it was not recognized as being very useful there, because the tracking application was not considered.

The major problem of tracking algorithms like the one proposed in [Lucas and Kanade, 1981] is that tracks are lost after some time. Therefore stabilization is a major issue. One interesting idea, which promises stable matches even over long baselines, has been proposed by Molton *et al.* [2004]. An algorithm for tracking point features over long image sequences is discussed that explicitly uses the local planarity assumption, which is only implicitly contained in [Lucas and Kanade, 1981]. The idea is to estimate the surface normal at the reconstructed 3d point and use this information for the matching. Conversely the estimate of the normal is updated along the sequence, enabling stable wide-baseline matches. Interestingly, this approach uses geometric cues in order to obtain better real-time tracking results strengthening again the vital importance of geometric information.

A graph theoretic view on the tracking problem is taken by Shafique and Shah [2005]. The problem of finding point tracks across image sequences is formulated in a graph theoretic

setting. While finding matches between two images is formulated as bipartite matching problem, which is solvable in polynomial time, the multi-view matching problem is NP-hard. This gives some insight into the inherent complexity of the task, although a greedy algorithm is proposed that promises real-time on-line tracking of image points over monocular sequences. The practical applicability, especially in comparison to the approach proposed by Molton *et al.* [2004], is not clear, though the problem of loosing tracks is clearly a consequence of the greedy strategy also used in [Lucas and Kanade, 1981] and [Molton *et al.*, 2004].

The point matching algorithms presented up to now were all based on comparing image patches between the images. In contrast to those comparison based approaches, there exist descriptor based approaches that compute for each interest point a single descriptor and use the distance between those descriptors for the matching. The descriptors must be designed, such that they are invariant against the anticipated image transformations. There are two advantages of this strategy: first, the descriptor needs only to be computed once for each feature, so that the time complexity is only linear in the number of features in contrast to the quadratic complexity of the comparison based methods. Second, descriptors can be stored for each image, so that large image databases are possible with this technique. For the task of matching, descriptor based and comparison based methods are equally applicable, though. Hence, some previous work on those techniques will be presented in the following.

Today the most prominent among the descriptor based approaches, which recently has received a lot of attention from the computer vision community due to its good performance, has been presented by Lowe [1999; 2004], who proposed an algorithm for matching interest points based on a scale invariant descriptor for each point, which is now commonly known under the acronym SIFT (scale invariant feature transformation). It is proposed to first extract interest points and their dominant rotation and scale by finding local extrema in scale space (cf. Lindeberg, [1998b]. For each such point in scale space a local image descriptor is computed by taking the histogram of gradient directions in the vicinity of the point. It is conjectured that Euclidean distances of this descriptors are very distinctive and useful for the task of database retrieval. This conjecture has also been confirmed by the study of Mikolajczyk and Schmid [2003] and this approach must be considered as most competitive for the task of purely radiometric point matching today. It could be used to supplement the geometric approach presented in section 3.3, but this path has not been followed yet.

Skrypnyk and Lowe [2004] report on a system for marker-less augmented reality based on SIFT descriptors. In an off-line initialization phase a metric 3d reconstruction of the scene is constructed from multi-view point matches based on SIFT feature descriptors (cf. Lowe, [2004]). For each 3d point the SIFT feature descriptor is retained, so that interest points from later images can be matched and their orientation can be computed.

The SIFT descriptor of Lowe [1999; 2004] is invariant to scale and rotation. Other invariances might be desirable. For instance assuming a locally planar surface and arbitrary viewpoints requires the descriptor to be affine invariant. Such an affine invariant descriptor has been proposed by Schaffalitzky and Zisserman [2001], who used the local second-moment-matrix in order to compute a normalized frame to compensate the effects of affine distortion (cf. Baumberg, [2000]). This normalized frame itself is used as descriptor and the residuals of a least-squares estimation of contrast- and brightness-difference are used as a measure of dissimilarity like in [Georgescu and Meer, 2004], which has been discussed above.

Other descriptors have been proposed in the literature. For instance Schmid and Mohr [1997] used the differential invariants of Koenderink and van Doorn [1987] computed for

automatically extracted interest points for the task of image retrieval. The retrieval results suggested that those descriptors are well suited for the task of feature matching, although in the more recent study by Mikolajczyk and Schmid [2003] they were outperformed by the SIFT descriptors of Lowe [1999; 2004]. The reason might be the fact that the descriptor is not scale invariant.

Very similar to the descriptor of Schmid and Mohr [1997] and also not scale invariant is the descriptor presented by Gouet *et al.* [1998], who devised a feature point descriptor specifically designed for color images. For each interest point a feature vector comprising of the local color and the local gradient of the color is computed. Matching is performed by looking at the residuals of a least squares estimation, which compensates the effects of some color and brightness changes like in [Schaffalitzky and Zisserman, 2001] and [Georgescu and Meer, 2004].

The major problem of invariances is that the discriminative power decreases. Vedaldi and Soatto [2005] point out that most feature detectors and descriptors implicitly assume a planar scene. It is shown that it is possible to construct viewpoint-invariant descriptors, but that those unfortunately cannot be shape-discriminative. It is also proved that discriminating scenes of different shape but identical albedo requires a reconstruction of the scene in the matching process. This rather theoretical results again strengthens the vital importance of incorporating geometric cues into the matching process, especially if certain invariances are desired.

A conceptually very different path is taken by Meltzer *et al.* [2004b], who apply machine learning techniques to learn the appropriate invariances from training data. The idea is to learn good feature descriptors from given image sequences by applying kernel principal component analysis on the regions around the interest points. With every new matched point the kernel principal components are learned incrementally and used for further matching. In [Meltzer *et al.*, 2004a] a simultaneous localization and mapping system using this descriptor is presented. The drawback of this approach is that the invariances are not made explicit. Furthermore, such a procedure has to be initialized with good initial training data, which has to be obtained by other matching methods.

This section reviewed some previous work on point feature matching. Two approaches were distinguished, namely comparison-based and descriptor-based methods. However, for the task of matching, in contrast to the task of database retrieval, the difference between those two methods is not relevant apart from performance issues. Furthermore two interrelated influence factors have been identified, namely the length of the baseline and the invariances of the image transformation. While radiometric methods usually perform better for small baselines, the geometric matching framework presented in section 3.3 is only able to work with long baselines, because the geometry is more stable in this case. The wide baseline methods require that the radiometric methods allow many invariances for the transformation between the images. This invariances weaken the discriminative power of such radiometric methods. Hence, a synthesis between the geometric and radiometric approaches is required. It has been seen that the geometric framework presented in section 3.3 easily allows to integrate many of the radiometric matching methods proposed in the literature.

Lines

Matching image lines has been by far not as popular as matching image points due to the weak image content in the vicinity of lines in images. Many line matching algorithms are extensions of ideas adopted from the point matching procedures discussed in the previous section. Like in the case of point matching, tracking line segments has received some attention and some prior work on this topic will be reviewed first. Also wide-baseline techniques and descriptor based methods have been proposed for matching line segments and will be discussed thereafter. Finally some previous work on line matching based on geometric cues will be presented, as those geometric cues are much more important for the task of line matching than for the task of point matching, and have therefore been used much more consequently there.

As in the point case, the geometric grouping framework of section 3.3 is not applicable in the case of short baselines. However, the line tracking might be a potential competing technique, if short baseline image sequences are available.

For the task of tracking line segments, the classical work of Lucas and Kanade [1981] has been extended toward line segments by Chiba and Kanade [1998]. They proposed a method for matching lines across two consecutive short-baseline frames by using the motion estimation of [Lucas and Kanade, 1981] to generate a motion prediction for each image line. The matches are scored according to image gradient directions and mutual overlap yielding the final matches.

Also Zhang [1994] uses such a prediction strategy. He proposed to use an extended Kalman-filter for tracking tokens, especially points and line segments, over an image sequence using the statistical as well as geometric properties of the entities. It is proposed to resolve matching ambiguities by retaining multiple and possibly contradicting matches and evaluate them based on their uncertainty measures. Both of this methods use geometric rather than radiometric cues.

The idea of Pritchett and Zisserman [1998b] has also been extended toward line segments. Beardsley *et al.* [1996] proposed to track points and lines across a sequence of three short baseline images applying a RANSAC approach. Therefore putative correspondences are generated based on very simple distance measures and, using the trifocal tensor as underlying model, the matching is established based on the geometry using a RANSAC procedure.

Only the work of Chiba and Kanade [1998] uses the radiometric image content in the vicinity of the line segment directly, while the other two approaches use only the mere existence of line segments at a certain location. This is basically the same kind of information used by the grouping framework presented here, with the difference that in the tracking case short baselines are assumed, while here the image orientation is assumed to be known.

Also wide-baseline techniques have been proposed for lines. Unlike the point case, most of this approaches assume the image geometry to be known somehow. For instance a comparison based wide baseline line matching algorithm for images with known epipolar geometry has been presented by Schmid and Zisserman [1997]. The idea is to assume a locally planar object in the vicinity of the scene line. Then the homography between the images of the line is defined up to a single rotation parameter for the scene plane. The cross-correlations of points in the vicinity of the line transfered by this family of homographies are computed and the final matching score is obtained by taking their maximum. In [Schmid and Zisserman,

2000] this idea to use a plane induced homography is extended from lines toward conics and smooth curves. This is one of the most promising approaches to be integrated as radiometric information, if the geometric grouping framework should be applied. Again, this path of integrating radiometric information has not been followed here.

Another such approach based on least squares matching has been presented by Cheng *et al.* [2001], who proposed a method for matching points across two and lines across three views. First for every pair of interest points the 3d point closest to the two projection rays and for every triple of interest lines the 3d line closest to the three projection planes are computed. This 3d entities are then back-projected into the images an a least-squares score is computed from the intensity data. This approach differs from the approach of Schmid and Zisserman [1997], as here in addition to using least-squares instead of cross-correlation the orientation of the scene plane is fixed and assumed to be aligned with the image planes.

As in the point case, also descriptor based matching techniques exist for lines. For instance Mikolajczyk *et al.* [2003] generalized the scale invariant feature point descriptor of Lowe [1999; 2004] toward edges. The histogram of the weighted gradient directions in the vicinity of the edges relative to the direction of the edge itself are used as a descriptor and matching is performed based on the distance of those histograms. The information contained in the vicinity of edges in images is quite weak, however, so that it must be strongly recommended to stabilize such a matching strategy with geometric cues. As in the point case, such radiometric distances are straightforwardly integrated into the geometric grouping framework presented in section 3.3. This path has not been followed, though, as the main goal of chapter 4 is to demonstrate the power of the geometric grouping approach without disturbance from radiometric image cues.

Instead of histograms of gradient directions other descriptors have been proposed. For instance Bay *et al.* [2005] proposed an algorithm for matching line segments between two images by establishing initial matches based on the distance of the color histogram of the lines neighboring pixels. This initial matches are then refined based on topological constraints. Also color histograms suffer the problem of a weak discriminative power, so that it would be interesting, how much is achieved by the topological constraints and how much is actually achieved by the distance measure.

As has already been seen in the previous paragraphs, information such as topology and geometry is a much stronger cue in line matching than the radiometric image content itself, as the content in the vicinity of edges is usually not very discriminative. This leads to the extreme idea of not using the radiometric image content for line matching at all, as will also be done in chapter 4. The reason here is to demonstrate the power of the geometric grouping framework without any disturbing influences from other factors. This matching based on the grouping framework presented in section 3.3 has been published in [Beder, 2004b].

Other authors have published line matching algorithms, which are solely based on geometric information, too. For instance Jung and Paparoditis [2003] presented a purely geometric algorithm for matching edgels across multiple oriented views. The idea is very similar to the graph theoretic grouping approach presented in section 3.4.2 and goes as follows: Every epipolar compatible pair of potentially matching edgels yields a 3d edgel hypothesis in space. Those edgel hypothesis are then grouped together yielding the final 3d reconstruction. Similar results to those in chapter 4 are obtained with the obvious difference that edgels instead

of points and lines are matched.

It has been seen in this section that line matching is much more reliant on image geometry than point matching. This could either be a short-baseline assumption, with some work on this topic presented in the first part of the section, or the known geometry derived for instance from point matches beforehand. As the discriminative power of radiometric cues in the vicinity of image lines is weak, the geometric framework presented in section 3.3 presents a true alternative to the radiometry based matching methods by itself. However, like in the point case both techniques are supplementary and best results can be expected from combining them, which is again easily possible.

Regions

Matching regions is much more complicated than matching points or lines between images due to the unknown scene structure corresponding to the depicted regions. This is because the problem of matching image regions can be considered as part of the dense (rather than feature based) matching and reconstruction problem. As mentioned above, those dense techniques will not be considered in this work but a comprehensive overview on the current state of the art in this field can be found in [Scharstein and Szeliski, 2002].

However, some region matching methods employ grouping techniques and will therefore be discussed in the following, as the proposed geometric grouping methodology can be applied to solve those problems. For instance Bartoli [2001] proposed a method for the fully automatic reconstruction of piecewise planar scenes from oriented images. The point matching, and hence a set of 3d points, is assumed to be given. Co-planar points are then grouped together using a RANSAC procedure and the generated plane hypothesis are verified using the radiometric information from the images. Except for the radiometric verification step, the proposed algorithm is very similar to the grouping framework presented in section 3.3. Due to the usually low redundancy contained in 3d point clouds obtained from sparse feature matching, the radiometric verification step is vital in this case. Hence, no such application is presented in section 4, as purely geometric cues turned out to be not sufficient in this case. This might be different, if more redundant datasets such as dense 3d point clouds obtained for instance using laser scanners are available. This has not been exploited here, though.

A very similar approach has also been proposed by Dick *et al.* [2000], who presented a method for the fully automatic reconstruction of planar scenes from multiple images. First interest points are extracted and matched using cross-correlation yielding a set of reconstructed 3d points. Like in [Bartoli, 2001], initial planes are found in this point cloud using a RANSAC procedure. The final parameters of the planar model are then refined using a probability model based on the spatial, boundary, texture and parallax parameters of the planes. Again, grouping points is only used to find initial plane hypothesis.

Region matching based on grouping co-planar 3d lines, rather than 3d points, has been proposed by Bignone *et al.* [1996], who presented a method for the fully automatic extraction of house roofs from oriented aerial images. First, edges are extracted in the most nadir image and the two-parameter family of corresponding 3d line segments is checked in all the other images based on geometric and radiometric similarity measures. The reconstructed 3d line segments are then grouped together based on co-planarity using a RANSAC procedure. This sub-task could also be easily formulated in the uncertain oriented projective geometric grouping framework presented in section 3.3, where some improvement might be achieved

by the rigorous statistical modeling. In comparison to the 3d point case considered in [Dick *et al.*, 2000] and [Bartoli, 2001], the co-planarity constraint imposed by 3d lines is much stronger, so that grouping those 3d lines might be more successful even in the case of low redundancy. However, this has not studied.

Also Baillard and Zisserman [1999; 2000] proposed a region matching algorithm based on grouping. In contrast to Bignone *et al.* [1996], their approach is based on grouping co-planar 3d planes rather than 3d lines. It is proposed to first generate 3d line hypothesis from pairs of extracted image line segments and then, following the idea of Schmid and Zisserman [1997], estimate the adjacent half planes by finding the plane angle, which maximizes the correlation between all images. Finally those half-planes are grouped based on co-planarity. This latter sub-task is again easily formulated in the uncertain oriented projective geometric grouping framework presented in section 3.3, where some improvement could be achieved by the rigorous statistical modeling. This has not yet been exploited, though.

It has been seen in this section that various proposals for solving the region matching problem contain grouping sub-tasks. Although the general dense matching problem is not the topic of this work, applying the grouping methodology presented in section 3.3 might offer improvements to some of those techniques. Again it must be stressed that the geometric grouping and radiometric verification in the images are supplementary and should be applied in combination in practical application, which has been seen to be true for all the discussed matching techniques.

1.4.2 Previous work on building reconstruction

In the previous section various approaches for feature extraction and feature matching have been presented. The ultimate goal of photogrammetry is to use this tools to reconstruct and measure the scene geometry depicted in an image. A subtask of this is the task of building reconstruction that is crucial for todays mapping applications and unfortunately not satisfactory automated so far. In chapter 4 of this work it will present a testbed for the proposed geometric grouping framework. Note that the results shown in chapter 4 are not meant to be used in practical applications, but rather show the power of the geometric framework, as they do not use radiometric information from the images at all. As already discussed in the previous section, certainly also radiometric cues from the images should be taken into account for any practical application. However, the results shown in chapter 4 suggest that ignoring geometric cues is not a good thing to do either, but a combination of both techniques should be applied. In the following an overview on some previous work regarding the task of building reconstruction from aerial images will be given.

First some overview material on the topic of building reconstruction will be presented. Much work on the topic can be found in the workshop proceedings [Grün *et al.*, 1995], [Grün *et al.*, 1997] and [Baltsavias *et al.*, 2001] as well as [Förstner and Plümer, 1997], where many research groups working in this field gave an overview on their various approaches.

A comprehensive overview on the field has been given by Mayer [1999], who reviewed the state of the art in automatic building reconstruction in this article. The different methods are assessed based on the building model used and the algorithmic strategy.

A more recent overview can be found in [Baltsavias, 2004]. It is pointed out that

knowledge-based and semi-automatic systems are currently the most promising approaches, since reliability and completeness together with automated evaluation poses the major problems. The need for reasoning in 3d as early as possible is formulated, because most knowledge is often expressed in this space. This is also propagated here in chapter 4.

Also Förstner [1999] gave a brief overview on the state of the art of automatic building reconstruction and its relation to semi-automatic methods. The complexity of building models makes fully automatic methods too unreliable for current practical demands, so that semi-automatic methods are still the only practical reasonable choice.

Mayer [2004] discusses the commercial applicability of current automatic object extraction methods. It is pointed out that the current state of the art is not yet sufficient for practical purposes. It is advocated that only on a rigorous theoretical basis, in particular statistical modeling, advances in this increasingly important field are achievable. This thesis constitutes a step in this direction of applying rigorous statistical methods to the tasks of geometric grouping and reconstruction.

Also Paparoditis *et al.* [1998] give an overview on current automatic building extraction methods from oriented stereo pairs and conclude that the current available methods are not sufficient for solving the task in a fully automatic fashion.

Most building extraction approaches are distinct from generic reconstruction algorithms by often being highly model driven. Braun *et al.* [1994] discussed the various types of models adequate for the task of building reconstruction. The models considered in the work are polyhedral, parameterized building primitives, boundary representation CAD and generic volumetric CSG. It is pointed out that there exists an aggregation hierarchy in object and in image space, and the mutual relations are discussed, as well as some internal constraints. The results presented in chapter 4 are therefore not typical for building reconstruction applications, as the only model assumption made is that building boundaries are straight line segments. This is a very weak assumption, as, in comparison to polyhedral models, no topological constraints are enforced. A post-processing might yield much better results, if such more restrictive and simplifying building models are enforced. In the following some previous work on building reconstruction employing increasingly general building models will be reviewed.

Cuboids Probably the most simple building model is to assume the buildings to be shaped like a cuboid, which is only the case for rectilinear flat-roof buildings. This model has for instance been used by Noronha and Nevatia [1997; 2001], who described an approach for the automatic reconstruction of rectangular buildings from multiple aerial images. Line segments are extracted and first grouped together into junctions. The resulting junctions are then grouped together into parallelograms, which are finally matched across the views using the known epipolar geometry and approximate ground hight. Note that grouping plays an important role in this system, although all grouping is performed in the 2d image domain rather than the scene domain as proposed here in chapter 4.

The model assumption of rectilinear flat-roofed buildings is so strong that Lin and Nevatia [1995; 1996; 1998] claim to be able to reconstruct such buildings from only one single aerial image using simple additional assumptions on shadows and image acquisition time. Like in [Noronha and Nevatia, 1997; 2001] features are extracted, grouped and building hypothesis in the image are generated and verified. Shadows and the known position of the sun are finally used to extract 3d information.

Another building reconstruction system, which is restricted to rectilinear flat-roofed build-

ings, has been presented by Collins *et al.* [1998]. In there edges are extracted and rectangular structures are found. Afterward this structures are matched across several views to generate 3d building hypothesis, which are finally refined using the intensity information from all images. It is pointed out that the difficulties of automatic building reconstruction might be overcome by incorporating a large number of views instead of devising increasingly elaborate matching strategies. This concurs with the findings of chapter 4, as the geometric grouping framework performs best in the presence of highly redundant datasets.

A cuboid building model is also assumed by Dick *et al.* [2001], who presented an approach for the reconstruction of architectural scenes from oriented close-range images. Following a metric reconstruction the three dominant planes in the scene are found using a RANSAC approach. After this basic cuboid has been found, the model is refined by detecting different parameterized parts, such as windows or doors, using a probabilistic model with shape and texture priors. The estimated parameters of the detected model parts are used thereafter to describe the scene more accurately. In [Dick *et al.*, 2002; 2004] this approach is augmented with a implicit global priors reflecting regularities in the building structure. The MAP estimates for the parameters are found using a MCMC scheme.

Basically the same model assumption has also been made by Werner *et al.* [2001; 2002b], who also presented an algorithm for building reconstruction from close-range images. The difference is that first the three vanishing lines of the three dominant planes occurring in architectural scenes are extracted in the image. Using the plane-sweep algorithm of Baillard and Zisserman [1999; 2000] on this lines, the three dominant planes are found. In [Werner and Zisserman, 2002a] the cuboid building model is further augmented with windows using the model based approach of [Dick *et al.*, 2001].

Parametric models One possible generalization of the cuboid image model is to allow a certain amount of pre-defined parametric building models instead of just one single model. This has been presented for instance by Fischer *et al.* [1998], who proposed a model-based approach for automatic building reconstruction from aerial images. Matched 2d corners are used to produce initial 3d corner hypothesis. This 3d corner hypothesis are then used to generate building hypothesis using a database of building models. Finally this 3d building hypothesis are verified against the image data. This latter step of generating and verifying model driven building hypothesis could be applied as a post-processing step for the approach presented in chapter 4 in order to obtain semantically meaningful building models.

A very similar idea has been presented by Spreeuwers *et al.* [1997], who proposed a model driven algorithm for building reconstruction from oriented aerial images. A 3d reconstruction of the scene is computed by verification of building primitive hypothesis and estimation of their parameters from multiple images. It is pointed out that the performance of the system significantly increases with the use of multiple images. This is in complete accordance with the findings in chapter 4, as the geometric grouping framework presented here is also reliant on highly redundant datasets.

Polyhedrons A major drawback of parametric building models is the restriction to a small set of pre-defined building models. As a solution polyhedral models have been proposed, which are in principle able to approximate any building structure. For instance Taylor *et al.* [1996] report on the well-known façade-system for polyhedral building reconstruction. The system works by matching edges of polyhedral primitives with edges in the images. Constraints on the mutual relation between the building primitives are then applied in order to estimate the polyhedral models. However, the system must be considered semi-automatic, as it relies on a lot of user interaction due to the many ambiguities inherently contain in the task of polyhedral reconstruction. As pointed out before, polyhedral reconstruction from

oriented images must be still considered an unsolved problem so far and also the geometric grouping results presented in chapter 4 must be seen as only being a pre-rpocessing step for subsequent polyhedral reconstruction.

An earlier work on automatic building reconstruction has been presented by Nevatia and Chung [1992], who discuss a methodology for the recovery of polyhedral building structures from a pair of aerial images. A hierarchy of image descriptions is built and matched at different levels. Matching constraints for the various features at the various levels are discussed.

Also Moons *et al.* [1998] described a workflow for the fully automatic polygonal roof reconstruction from aerial images. First 3d scene line hypothesis from the longest extracted image line segments are generated using the trifocal constraint. Those 3d lines are then grouped together based on co-planarity and, including smaller 3d line segments, 3d polygon hypothesis are generated. Thereafter, starting form the convex hull, concave structures are found by looking at small line segments having just one end-point on the convex hull. The remaining boundary line segments are classified according to their angle with the ground plane and from this classification and the neighborhood structure a wire-frame roof-model is generated. Finally this model is adjusted by back-projecting it into the original images. This approach is also reliant on grouping geometric primitives, hence the proposed uncertain geometric grouping framework might be applicable for improving certain sub-tasks.

Another system for the automatic extraction of polyhedral building reconstruction has been reported by Henricsson [1997; 1998]. The system is based on the roof extraction algorithms described in [Bignone *et al.*, 1996]. The achieved accuracy measured against a reference data set is analyzed. It is pointed out in [Henricsson, 1998] that color cues are important for obtaining reliable results and that a pure geometric approach has to be augmented with such radiometric information in order to work.

A system for the reconstruction of buildings from terrestrial stereoscopic image sequences, rather than oriented aerial images, has been presented by Koch [1993; 1994a; 1994b; 1995] and Liedtke [1993], which is based on computing depth maps for each stereo pair by correlation. Those depth maps are then segmented and the segments are triangulated. The motion along the sequence is recovered and the triangulated surfaces are registered to build a 3d model. In [Koch, 1996] a polygonal model is created by fitting planar surfaces to the meshes. In [Koch *et al.*, 1998a; 1998b; 1999a; 2000; Koch and Frahm, 2001] a self-calibrated monocular sequence is used instead of a stereo rig.

Planes In contrast to requiring the reconstructed objects to be complete polyhedrons, some work only reconstructs dominant structures such as isolated planes, lines or points. Although those models are more complicated to use than the ones presented in the previous paragraphs, they are much easier to compute, as no restrictions on the mutual relations between the primitives need to be enforced. The building reconstruction application presented in chapter 4 also falls in this category of applying feature based reconstruction algorithms to the task of building reconstruction from aerial images.

This has also been done by Baillard *et al.* [1999], who report on the application of the plane-sweep matching techniques presented in [Baillard and Zisserman, 1999] to the task of building reconstruction from aerial images. No specific prior knowledge on the structure of buildings apart from the assumption of planar surfaces are made in this approach, but the matching technique is merely applied to aerial images. This is also done in chapter 4, as there also no building specific assumptions are enforced.

Many building reconstruction applications do not rely only on images, but incorporate

data acquired with other sensors as well. The geometric grouping framework presented in section 3.3 is in principle not restricted to a certain sensor type and the possibility of straightforward sensor fusion is another major advantage of the presented framework. Hence, some prior work on building reconstruction using additional sensors will be presented.

The most common sensor apart from aerial images for the task of building reconstruction is the airborne laser scanner. This has for instance been used as an additional sensor by Jaynes *et al.* [1997], who proposed to improve the toolbox presented by Collins *et al.* [1998] to cope with more complex buildings by incorporating registered range data in order to classify the observed roofs based on a more complex 3d building primitive database.

Also Kim *et al.* [2001b; 2001a; 2002; 2004] proposed a model based building reconstruction method from aerial images and digital elevation data. First line features are extracted and nearby lines are grouped into junctions. The DEM is used to remove line features, which are not located near any building. Then 3d features are generated by pairwise matching the junctions and are again verified against the DEM. The remaining 3d features are grouped together and, using a building model of rectangular shaped gable roofs, building hypothesis are generated, which are finally verified using various cues. Note that this approach is mainly based on grouping image features and the elevation data is mainly used to reduce the number of outliers.

One major problem not considered by the works presented in this paragraph so far, which needs to solved for fusing data acquired using multiple sensors, is the registration of this data. This task is very difficult especially if the sensors are mounted on different platforms and operate on different scales. For the task of building reconstruction it is very common to geo-reference the data, but other approaches have been proposed. For instance Früh and Zakhor [2003] report on the generation of 3d city models by fusing airborne images and laser-scans with vehicle mounted images and laser-scans. A particle-filter based Monte-Carlo localization technique is proposed for registering the laser scans with each other. The 3d points obtained by the lase scanners are then grouped into 3d planes, which are finally verified against the images acquired simultaneously. The problem of simultaneously grouping 3d points with different accuracies resulting from the two different platforms they were acquired with is not addressed. Here the geometric grouping framework offers a rigorous treatment of such different uncertainties from which such sensor fusion techniques could benefit. This path has not been followed yet, though.

This section reviewed some previous work on building reconstruction. It has been seen that most of the works make very simplifying assumptions on the geometric structure of the buildings. This on the one hand eases the task significantly but is on the other hand much to restrictive to be applicable in practice. It has also been seen that the currently no fully automatic building reconstruction approach exists that would satisfy the practical demands. This holds also true for the results presented in chapter 4. However, the results presented there will demonstrate the strength of geometric cues for the task, even if a much weaker building model than in many of the prior work is assumed. Furthermore the rigorous treatment of uncertainties in the presented framework allows the straightforward fusion of multiple sensors that has been identified as being very important for practical applications.

Chapter 2

Uncertain oriented projective geometry

This chapter describes the underlying concepts of uncertain oriented projective geometry. The first section is about the representation of such objects that will be used throughout this work. The second section describes methods for reasoning about such objects, i.e. making decisions about relations between the objects. Finally, the third section will be concerned with efficiency. A data structure will be presented that allows to perform the reasoning presented in the second section of this chapter in an efficient manner.

2.1 Representing uncertain oriented entities

In this section a representation for uncertain oriented projective entities will be presented. The concepts are very closely related to the ideas presented by Heuel [2004] for representing uncertain projective entities. The novelty lies in the use of oriented projective geometry throughout this work that allows to derive additional relations between the entities and also allows to represent novel, more complex compound uncertain entities. Among those novel uncertain compound entities are line segments in 2d and 3d, polygons in 2d and 3d as well as 2d edgels and 3d facets that are often very useful. In the following the underlying notions for representing uncertain oriented projective entities will be presented that are then specialized toward the specific base entities followed by the specific compound entities.

In an Euclidean setting, every uncertain entity is given by a probability density function

$$p(\boldsymbol{x}) : I\!R^n \mapsto I\!R^+ \tag{2.1}$$

so that the probability that the entity lies within a region $\Omega \subset I\!R^n$ is given by

$$\mu(\Omega) = \int_\Omega p(\boldsymbol{x}) d\boldsymbol{x} \tag{2.2}$$

Note that here distributions are assumed to be generalized functions in the sense defined by Horn [1986, p.116] to avoid problems with normalization and integrability, if they are only non-zero on a sub-set of the integration space. In many cases this distribution can be characterized sufficiently by its first and second moments

$$\boldsymbol{m}_x = \int \boldsymbol{x} p(\boldsymbol{x}) d\boldsymbol{x} \tag{2.3}$$

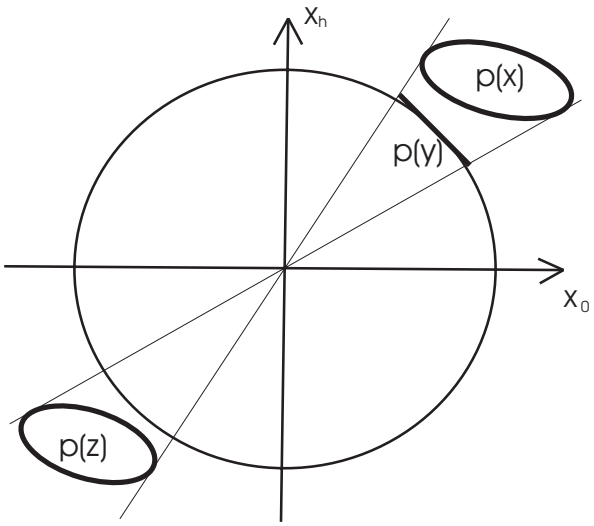

Figure 2.1: Uncertain oriented projective entities are represented by their probability distributions. The mean can be interpreted as a position and the covariance matrix as a confidence ellipsoid in oriented projective space. Here the entities $\mathbf{x}$ and $\mathbf{y}$ are identical, whereas the entity $\mathbf{z}$ is not in uncertain oriented projective geometry. Also note that the distribution of $\mathbf{y}$ is degenerate resulting from a singular covariance matrix of a spherical normalized entity.

$$\mathsf{C}_{xx} = \int (\boldsymbol{x} - \boldsymbol{m}_x)(\boldsymbol{x} - \boldsymbol{m}_x)^T p(\boldsymbol{x}) d\boldsymbol{x} \tag{2.4}$$

Using a maximum entropy approach, the opposite direction is possible, too. Namely knowing only those first two moments, the distribution having maximum entropy is given by the Normal distribution (cf. [Kagan *et al.*, 1973, p.410] and [Cover and Thomas, 1991, p.270])

$$\begin{aligned} p(\boldsymbol{x}) &= \operatorname{argmax}_p \left(- \int p(\boldsymbol{x}) \log p(\boldsymbol{x}) d\boldsymbol{x} \right) && (2.5) \\ &= \frac{1}{\sqrt{(2\pi)^n |\mathsf{C}_{xx}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_x)^T \mathsf{C}_{xx} (\boldsymbol{x}-\boldsymbol{m}_x)} && (2.6) \end{aligned}$$

Now a similar notion in an oriented projective setting is developed. In this setting, two vectors are considered to be equivalent, if and only if there exists a positive factor that relates them (cf. [Heuel, 2004, p.20])

$$\mathbf{x} \equiv \mathbf{y} \Leftrightarrow \exists \lambda > 0 : \boldsymbol{x} = \lambda \boldsymbol{y} \tag{2.7}$$

Throughout this work such homogeneous vectors will be denoted with upright letters $\mathbf{x}$ in contrast to non-homogeneous vectors denoted as $\boldsymbol{x}$ following the notation of [Mugnier *et al.*, 2004].

The above notion carries over to the uncertain entities as follows: again every uncertain entity is given by a probability density function

$$p(\mathbf{x}) : I\!R^{n+1} \mapsto I\!R^+ \tag{2.8}$$

Now in analogy to the above equality definition two densities are considered to represent the same uncertain entity, if the following holds (see figure 2.1)

$$p_x(\mathbf{x}) \equiv p_y(\mathbf{y}) \Leftrightarrow \forall \Omega \subset I\!R^{n+1} : \int_{\{\mathbf{x}|\mathbf{x}\equiv\mathbf{z},\mathbf{z}\in\Omega\}} p_x(\mathbf{x}) d\mathbf{x} = \int_{\{\mathbf{y}|\mathbf{y}\equiv\mathbf{z},\mathbf{z}\in\Omega\}} p_y(\mathbf{y}) d\mathbf{y} \tag{2.9}$$

From this follows that the probability of an entity lying in a region $\Omega \subset \mathbb{R}^{n+1}$ of oriented projective space is given by

$$\mu(\Omega) = \int_{\{\mathbf{x}|\mathbf{x}\equiv\mathbf{z},\mathbf{z}\in\Omega\}} p(\mathbf{x})d\mathbf{x} \tag{2.10}$$

Again the density function can be characterized by its first and second moments

$$\mathbf{m}_x = \int \mathbf{x}p(\mathbf{x})d\mathbf{x} \tag{2.11}$$

$$\mathsf{C}_{xx} = \int (\mathbf{x}-\mathbf{m}_x)(\mathbf{x}-\mathbf{m}_x)^T p(\mathbf{x})d\mathbf{x} \tag{2.12}$$

Note that covariance matrices of such homogeneous entities will be denoted with upright letters C_{xx} in contrast to covariance matrices of non-homogeneous entities denoted with C_{xx}.

As many probability density functions represent the same uncertain entity, this characterization is not unique. To cope with this problem of comparability, a normalization to spherical coordinates is useful, though not required for every application. The idea is as follows: since multiplication with a positive scalar does not change the entity, the distribution is projected onto the tangent space of the unit sphere at the projection of its mean. Using linear error propagation (cf. [Koch, 1997, p.108]), the moments change according to (cf. [Heuel, 2004, p.110])

$$\mathbf{m}_x^{(s)} = \frac{\mathbf{m}_x}{|\mathbf{m}_x|} \tag{2.13}$$

$$\mathsf{C}_{xx}^{(s)} = \mathsf{J}^{(s)}\mathsf{C}_{xx}\mathsf{J}^{(s)^T} \tag{2.14}$$

with the Jacobian

$$\mathsf{J}^{(s)} = \frac{\partial \mathbf{m}_x^{(s)}}{\partial \mathbf{m}_x} = \frac{1}{|\mathbf{m}_x|}\left(I_d - \frac{\mathbf{m}_x\mathbf{m}_x^T}{|\mathbf{m}_x|^2}\right) \tag{2.15}$$

This representation for uncertain entities in oriented projective space is unique. Note that the covariance matrix $\mathsf{C}_{xx}^{(s)}$ is singular due to the projection on the tangent space. Again the interpretation of those moments as probability density function is also possible, because the maximum entropy distribution inside the tangent space is again the Normal distribution. Therefore the uncertain entities, the moments and the probability density functions will not be distinguished in the following. The usual notation for an uncertain entity will be the tuple $(\mathbf{m}_x, \mathsf{C}_{xx})$ containing the first two moments of its probability density function.

The next required generalization of the concept is toward compound entities, i.e. entities that comprise of a fixed number of oriented projective entities $\mathbf{x}_1, ..., \mathbf{x}_N$. In analogy to the above definition (cf. equation (2.7)), compound oriented projective entities are considered to be equivalent, if and only if there exist positive factors that relate them

$$(\mathbf{x}_1, ..., \mathbf{x}_N) \equiv (\mathbf{y}_1, ..., \mathbf{y}_N) \Leftrightarrow \mathbf{x}_1 \equiv \mathbf{y}_1 \wedge ... \wedge \mathbf{x}_N \equiv \mathbf{y}_N \tag{2.16}$$

Again this carries over to uncertain compound oriented projective entities defined as probability density functions

$$p(\mathbf{x}_1, ..., \mathbf{x}_N) : \mathbb{R}^{n_1+1} \times \cdots \times \mathbb{R}^{n_N+1} \mapsto \mathbb{R}^+ \tag{2.17}$$

that represent the same uncertain compound entity, if the following holds

$$p_x(\mathbf{x}_1, ..., \mathbf{x}_N) \equiv p_y(\mathbf{y}_1, ..., \mathbf{y}_N) \tag{2.18}$$

$$\Leftrightarrow \forall \Omega_1 \times \cdots \times \Omega_N \subset I\!R^{n_1+1} \times \cdots \times I\!R^{n_N+1} :$$
$$\int_{\{\mathbf{x}_1|\mathbf{x}_1 \equiv \mathbf{z}_1, \mathbf{z}_1 \in \Omega_1\}} \cdots \int_{\{\mathbf{x}_N|\mathbf{x}_N \equiv \mathbf{z}_N, \mathbf{z}_N \in \Omega_N\}} p_x(\mathbf{x}_1, ..., \mathbf{x}_N) d\mathbf{x}_1 \cdots d\mathbf{x}_N$$
$$= \int_{\{\mathbf{y}_1|\mathbf{y}_1 \equiv \mathbf{z}_1, \mathbf{z}_1 \in \Omega_1\}} \cdots \int_{\{\mathbf{y}_N|\mathbf{y}_N \equiv \mathbf{z}_N, \mathbf{z}_N \in \Omega_N\}} p_y(\mathbf{y}_1, ..., \mathbf{y}_N) d\mathbf{y}_1 \cdots d\mathbf{y}_N$$

Hence, the probability of a compound entity lying in a region $\Omega_1 \times \cdots \times \Omega_N \subset I\!R^{n_1+1} \times \cdots \times I\!R^{n_N+1}$ of the compound oriented projective space is given by

$$\mu(\Omega_1 \times \cdots \times \Omega_N) = \int_{\{\mathbf{x}_1|\mathbf{x}_1 \equiv \mathbf{z}_1, \mathbf{z}_1 \in \Omega_1\}} \cdots \int_{\{\mathbf{x}_N|\mathbf{x}_N \equiv \mathbf{z}_N, \mathbf{z}_N \in \Omega_N\}} p(\mathbf{x}_1, ..., \mathbf{x}_N) d\mathbf{x}_1 \cdots d\mathbf{x}_N \tag{2.19}$$

Again, the density will be characterized by its first two moments and vice versa. Thus the means and covariances are

$$\mathbf{m}_{x_i} = \int \mathbf{x}_i p(\mathbf{x}_1, ..., \mathbf{x}_N) d\mathbf{x}_1 \cdots d\mathbf{x}_N \tag{2.20}$$

$$\mathsf{C}_{x_i x_j} = \int (\mathbf{x}_i - \mathbf{m}_i)(\mathbf{x}_j - \mathbf{m}_j)^T p(\mathbf{x}_1, ..., \mathbf{x}_N) d\mathbf{x}_1 \cdots d\mathbf{x}_N \tag{2.21}$$

and, as many density functions represent the same entity, uniqueness is again achievable by spherical normalization, which is defined for compound entities as

$$\mathbf{m}_{x_i}^{(s)} = \frac{\mathbf{m}_{x_i}}{|\mathbf{m}_{x_i}|} \tag{2.22}$$

$$\mathsf{C}_{x_i x_j}^{(s)} = \mathsf{J}_i^{(s)} \mathsf{C}_{x_i x_j} \mathsf{J}_j^{(s)^T} \tag{2.23}$$

with the Jacobians for linear error propagation being

$$\mathsf{J}_i^{(s)} = \frac{\partial \mathbf{m}_{x_i}^{(s)}}{\partial \mathbf{m}_{x_i}} = \frac{1}{|\mathbf{m}_{x_i}|} \left(I_d - \frac{\mathbf{m}_{x_i} \mathbf{m}_{x_i}^T}{|\mathbf{m}_{x_i}|^2} \right) \tag{2.24}$$

The maximum entropy distribution for those first and second moments is the Normal distribution in the space spanned by the covariance matrices. Therefore also the uncertain compound entities will not be distinguished from their distributions or their moments and will be notated in the following as $(\{\mathbf{m}_{x_i}\}, \{\mathsf{C}_{x_i x_j}\})$, leaving out covariance matrices from the notation where they are not needed subsequently.

Note that the basic uncertain oriented projective entity is a special case of the compound uncertain oriented entity containing only one element.

Having now introduced the fundamental concepts underlying the representation of each uncertain oriented projective entity, in the following those notions will be specialized toward some useful specific geometric 2d and 3d entities. First, some base entities will be presented, which are then combined into some useful compound entities. The section is concluded by a representation for projective cameras in the uncertain oriented projective framework. In particular it will be shown explicitly for every transformation, how the uncertainties of the entities transform.

2.1.1 Uncertain oriented 2d base entities

As computer vision and photogrammetry is mainly concerned with processing images, 2d primitives such as points and lines naturally occur very often. In the following a representation of such entities in the uncertain oriented projective framework will be shown.

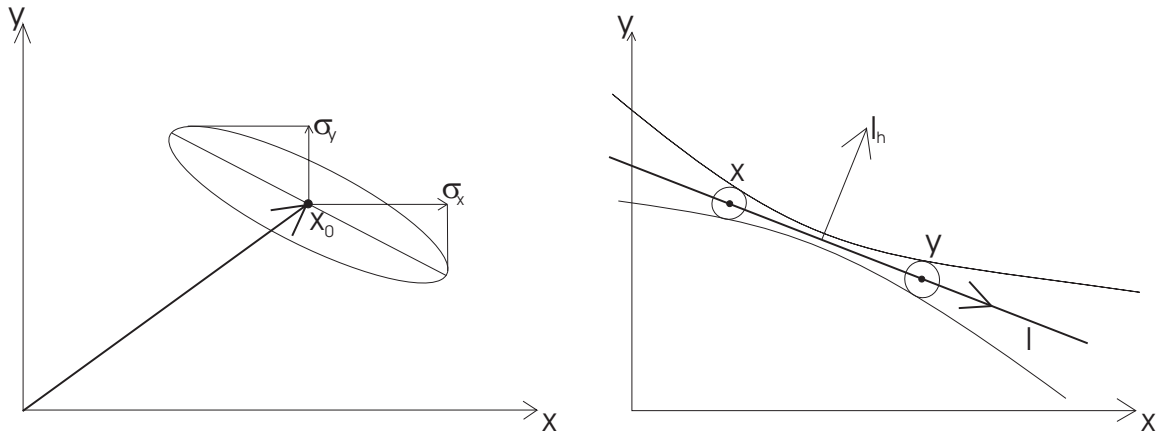

Figure 2.2: *Left:* An uncertain point in uncertain Euclidean space is represented by its position $\boldsymbol{x}_0$ and its uncertainties along the two axes σ_x and σ_y. The confidence ellipsoid needs not to be aligned with the coordinate system, but could also be rotated, which would result in a non-zero covariance σ_{xy}. *Right:* A directed line from the uncertain point $\mathbf{x}$ to the uncertain point $\mathbf{y}$. The error band is in this case a hyperbola rather than an ellipse. Also note that the normal of the line points by definition to the left seen in the direction of the line.

2d points

Starting with 2d points, it has been discussed in section 1.3.3 that there are many algorithms for extracting point features together with their uncertainties from images. Such a 2d point is then usually given by its Euclidean coordinates (x, y) together with their variances σ_x^2 and σ_y^2 and their covariance σ_{xy}. The uncertain Euclidean entity is thus represented as (see figure 2.2, left)

$$(\boldsymbol{x}, \mathsf{C}_{xx}) = \left(\begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix} \right) \tag{2.25}$$

The corresponding uncertain oriented projective entity is obtained as (cf. [Heuel, 2004, p.106])

$$(\mathbf{x}, \mathsf{C}_{xx}) = \left(\begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix}, \begin{pmatrix} \mathsf{C}_{xx} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} \right) \tag{2.26}$$

Given on the other hand an uncertain oriented projective 2d point

$$(\mathbf{x}, \mathsf{C}_{xx}) = \left(\begin{pmatrix} \boldsymbol{x}_0 \\ x_h \end{pmatrix}, \mathsf{C}_{xx} \right) \tag{2.27}$$

this may be normalized into Euclidean coordinates dividing by $|x_h|$ resulting in the normalized 2d point (cf. [Heuel, 2004, p.109f])

$$(\mathbf{x}^{(e)}, \mathsf{C}_{xx}^{(e)}) = \left(\frac{1}{|x_h|} \begin{pmatrix} \boldsymbol{x}_0 \\ x_h \end{pmatrix}, \mathsf{J}_e \mathsf{C}_{xx} \mathsf{J}_e^T \right) \tag{2.28}$$

$$= \left(\begin{pmatrix} \boldsymbol{x}_0^{(e)} \\ \operatorname{sign}(x_h) \end{pmatrix}, \begin{pmatrix} \mathsf{C}_{x_0 x_0}^{(e)} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} \right) \tag{2.29}$$

using the Jacobian

$$J_e = \frac{\partial \mathbf{x}^{(e)}}{\partial \mathbf{x}} = \frac{1}{|x_h|} \begin{pmatrix} I_2 & -\frac{\boldsymbol{x}_0}{|x_h|} \\ \mathbf{0}^T & 0 \end{pmatrix} \tag{2.30}$$

Observe that the orientation is preserved by this operation, so that an Euclidean point is only obtained, if the homogeneous coordinate is positive. Thus, it is always possible to represent a given uncertain Euclidean 2d point in the oriented projective framework, but not every uncertain oriented projective 2d point is representable as Euclidean point.

2d lines

A second useful feature encountered are straight 2d lines. Those 2d lines are usually created by connecting two 2d points. In the uncertain oriented projective framework this is done as follows: using the skew symmetric matrix inducing the cross product (cf. [Heuel, 2004, p.51])

$$S(\boldsymbol{x}) = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \tag{2.31}$$

the line from the point $(\mathbf{x}, \mathsf{C}_{xx})$ in the direction toward the point $(\mathbf{y}, \mathsf{C}_{yy})$ is defined as $(\mathbf{l}, \mathsf{C}_{ll})$ with (cf. [Heuel, 2004, p.51] and see figure 2.2, right)

$$\mathbf{l} = S(\mathbf{x})\mathbf{y} = -S(\mathbf{y})\mathbf{x} \tag{2.32}$$

and for uncorrelated $\mathbf{x}$ and $\mathbf{y}$

$$\mathsf{C}_{ll} = S(\mathbf{x})\mathsf{C}_{yy}S(\mathbf{x})^T + S(\mathbf{y})\mathsf{C}_{xx}S(\mathbf{y})^T \tag{2.33}$$

Observe that the vector $\mathbf{l}$ is bilinear in the vectors $\mathbf{x}$ and $\mathbf{y}$ and that changing the order of the two points changes the sign of the resulting connecting line. Again an Euclidean interpretation is possible, too. Given the uncertain oriented projective 2d line

$$(\mathbf{l}, \mathsf{C}_{ll}) = \left(\begin{pmatrix} \boldsymbol{l}_h \\ l_0 \end{pmatrix}, \mathsf{C}_{ll} \right) \tag{2.34}$$

the Euclidean entity is obtained dividing by $|\boldsymbol{l}_h|$ as (cf. [Heuel, 2004, p.109f])

$$(\mathbf{l}^{(e)}, \mathsf{C}_{ll}^{(e)}) = \left(\frac{1}{|\boldsymbol{l}_h|} \begin{pmatrix} \boldsymbol{l}_h \\ l_0 \end{pmatrix}, J_e \mathsf{C}_{ll} J_e^T \right) \tag{2.35}$$

$$= \left(\begin{pmatrix} \cos\phi \\ \sin\phi \\ -d \end{pmatrix}, \mathsf{C}_{ll}^{(e)} \right) \tag{2.36}$$

using the Jacobian

$$J_e = \frac{\partial \mathbf{l}^{(e)}}{\partial \mathbf{l}} = \frac{1}{|\boldsymbol{l}_h|} \begin{pmatrix} I_2 - \frac{\boldsymbol{l}_h \boldsymbol{l}_h^T}{|\boldsymbol{l}_h|^2} & \mathbf{0} \\ \frac{-l_0 \boldsymbol{l}_h^T}{|\boldsymbol{l}_h|^2} & 1 \end{pmatrix} \tag{2.37}$$

This yields the oriented angle of the line with the x-axis ϕ and the oriented distance to the origin, which is also known as the Hessian normal form.

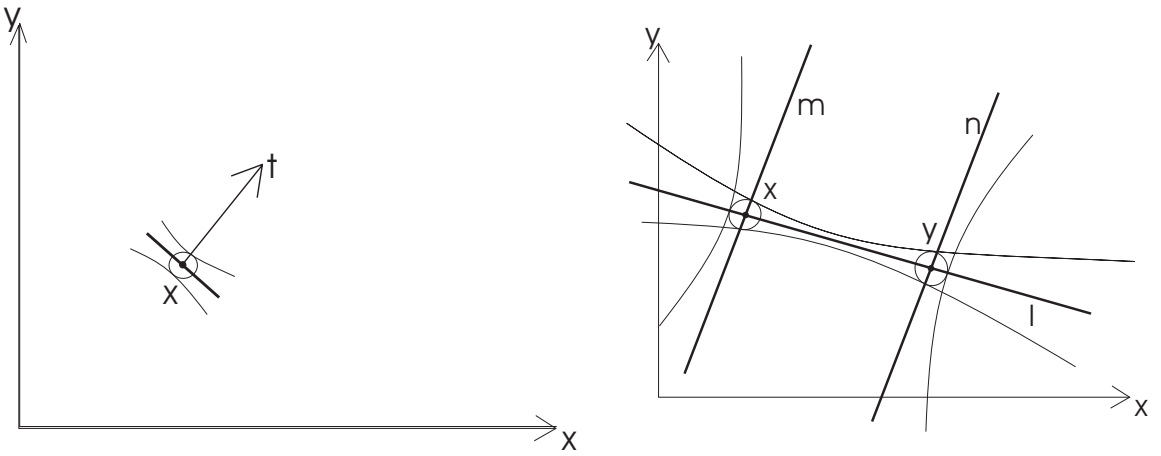

Figure 2.3: *Left:* An uncertain oriented projective 2d edgel located at the point $\mathbf{x}$ having the normal $\mathbf{t}$. *Right:* An uncertain oriented 2d line segment can be either represented by its two end-points $\mathbf{x}$ and $\mathbf{y}$ or by the connecting line $\mathbf{l}$ and the two delimiting lines $\mathbf{m}$ and $\mathbf{n}$. Observe that the shape of the confidence region of the line segment depends on the representation.

For duality reasons the intersection of two uncertain oriented projective lines $(\mathbf{l}, \mathsf{C}_{ll})$ and $(\mathbf{m}, \mathsf{C}_{mm})$ is defined as $(\mathbf{x}, \mathsf{C}_{xx})$ with (cf. [Heuel, 2004, p.54])

$$\mathbf{x} = \mathsf{S}(\mathbf{l})\mathbf{m} = -\mathsf{S}(\mathbf{m})\mathbf{l} \tag{2.38}$$

and its covariance matrix is given in the case of uncorrelated $\mathbf{l}$ and $\mathbf{m}$ by

$$\mathsf{C}_{xx} = \mathsf{S}(\mathbf{l})\mathsf{C}_{mm}\mathsf{S}(\mathbf{l})^T + \mathsf{S}(\mathbf{m})\mathsf{C}_{ll}\mathsf{S}(\mathbf{m})^T \tag{2.39}$$

The orientation of the resulting point depends on the oriented angle between the two lines. To see how, observe that for Euclidean normalized lines the last component of the resulting point is

$$x_3 = l_1 m_2 - l_2 m_1 = \cos\phi \sin\psi - \sin\phi \cos\psi = \sin(\psi - \phi) \tag{2.40}$$

which is positive for $\psi - \phi \in [0..\pi]$ and negative for $\psi - \phi \in [\pi..2\pi]$.

Up to now uncertain oriented projective 2d points and uncertain oriented projective 2d lines have been presented. Those two basic entities are now combinable into more complex compound entities, which will be shown in the following.

2.1.2 Uncertain oriented 2d compound entities

Now the 2d base entities presented in the previous section will be combined into more complex compound entities. It will first be shown, how 2d edgles can be represented followed by the representation of 2d line segments and concluded by 2d polygons.

2d edgels

The first compound entity is the 2d edgel, i.e. a position in 2d space with an associated direction. Its usefulness stems for instance from the fact that some feature extraction procedures yield not only the position of a feature point, but also its orientation. Those entities

are called edgels and consist of a position being an uncertain oriented 2d point $(\mathbf{x}, \mathsf{C}_{xx})$ and a normal

$$(\mathbf{t}, \mathsf{C}_{tt}) = \left(\begin{pmatrix} t_1 \\ t_2 \end{pmatrix}, \mathsf{C}_{tt} \right) \tag{2.41}$$

being an uncertain oriented projective 1d point representing a 2d direction usually aligned with the local gradient in the image. The resulting 2d edgel is then defined as the compound oriented projective entitiy (see figure 2.3, left)

$$(\{\mathbf{x}, \mathbf{t}\}, \{\mathsf{C}_{xx}, \mathsf{C}_{tt}, \mathsf{C}_{xt}\}) \tag{2.42}$$

The point at infinity corresponding to the normal direction vector is computable using

$$C_\infty = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{2.43}$$

as

$$\mathbf{x}_\infty = C_\infty \mathbf{t} \tag{2.44}$$

having the covariance matrix

$$\mathsf{C}_{x_\infty x_\infty} = C_\infty \mathsf{C}_{tt} C_\infty^T \tag{2.45}$$

Finally the line going though the edgel is constructible from the point $\mathbf{x}$ and the rotated point at infinity $\mathbf{x}_\infty$ by using the rotation matrix

$$R_\perp = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.46}$$

as

$$\mathbf{l} = S(\mathbf{x}) R_\perp \mathbf{x}_\infty = -S(R_\perp \mathbf{x}_\infty)\mathbf{x} \tag{2.47}$$

having the covariance matrix

$$\mathsf{C}_{ll} = S(\mathbf{x}) R_\perp \mathsf{C}_{x_\infty x_\infty} R_\perp^T S(\mathbf{x})^T + S(R_\perp \mathbf{x}_\infty) \mathsf{C}_{xx} S(R_\perp \mathbf{x}_\infty)^T \tag{2.48}$$

if the edgels normal is uncorrelated with its position, i. e. $\mathsf{C}_{xt} = \mathbf{0}$.

The additional information contained in the orientation of an edgel in comparison to just using the position of a feature point is often of great advantage for grouping tasks, as the search region can be significantly reduced.

2d line segments

Another important image feature extractable with the methods discussed in section 1.3.3 are straight line segments. Those line segments differ from straight lines by having a start- and an end-point. In an Euclidean setting, this means that line segments have a finite length. As will be seen, this is not the case in an oriented projective setting.

It will become clear in section 2.2.2 that two different representations for line segments are necessary to formulate various geometric relations. Therefore in the following two representations for uncertain line segments will be defined and it is shown, how those representations can be converted from one to the other.

Line segments in point representation: An uncertain 2d line segment is usually constructed from its two end-points. In contrast to the 2d line it does not extend from its

two end points in both directions, though it might cross the line at infinity and has therefore infinite length in an Euclidean interpretation. The most natural representation is given by the compound entity defined by the ordered pair of end-points $(\mathbf{x}, \mathsf{C}_{xx})$ and $(\mathbf{y}, \mathsf{C}_{yy})$ themselves (see figure 2.3, right)

$$(\{\mathbf{x}, \mathbf{y}\}, \{\mathsf{C}_{xx}, \mathsf{C}_{yy}, \mathsf{C}_{xy}\}) \tag{2.49}$$

The interpretation is as follows: starting from point $\mathbf{x}$ all points along the directed line $\mathbf{l} = \mathsf{S}(\mathbf{x})\mathbf{y}$ lie on the line segment until the point $\mathbf{y}$ is reached. The line at infinity is crossed, if and only if the orientation of the two end-points $\mathbf{x}$ and $\mathbf{y}$ is different.

Line segments in line representation: Another useful representation uses the connecting line $(\mathbf{l}, \mathsf{C}_{ll})$ between the end-points and the two delimiting lines $(\mathbf{m}, \mathsf{C}_{mm})$ and $(\mathbf{n}, \mathsf{C}_{nn})$ going perpendicular through the end-points (see figure 2.3, right)

$$(\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}, \{\mathsf{C}_{ll}, \mathsf{C}_{mm}, \mathsf{C}_{nn}, \mathsf{C}_{lm}, \mathsf{C}_{ln}, \mathsf{C}_{mn}\}) \tag{2.50}$$

Not all such entities actually represent a line segment, because of the perpendicularity constraint. The notion of orthogonality is defined using the conic that is dual to the two circular points (cf. [Hartley and Zisserman, 2000, p.33f])

$$C_\infty^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.51}$$

The perpendicularity constraints are then formulated as

$$\mathbf{l}^T C_\infty^* \mathbf{m} = 0 \tag{2.52}$$

and

$$\mathbf{l}^T C_\infty^* \mathbf{n} = 0 \tag{2.53}$$

As orthogonality is an Euclidean quantity, the end-points are not allowed to be at infinity in this representation.

Converting from point to line representation: To obtain a line segment in line representation, one first constructs the connecting line as shown above (cf. equation (2.32))

$$\mathbf{l} = \mathsf{S}(\mathbf{x})\mathbf{y} = -\mathsf{S}(\mathbf{y})\mathbf{x} \tag{2.54}$$

From the perpendicularity constraint (cf. equation (2.52)) follows that the ideal point

$$\mathbf{x}' = C_\infty^* \mathbf{l} \tag{2.55}$$

lies on $\mathbf{m}$. As the end-points are assumed not to be at infinity, it is distinct from $\mathbf{x}$. Therefore $\mathbf{m}$ is constructed by connecting $\mathbf{x}$ and $\mathbf{x}'$ yielding

$$\begin{aligned} \mathbf{m} &= \mathsf{S}(\mathbf{x})\mathbf{x}' && (2.56) \\ &= \mathsf{S}(\mathbf{x}) C_\infty^* \mathsf{S}(\mathbf{x})\mathbf{y} && (2.57) \\ &= \mathsf{U}(\mathbf{x})\mathbf{y} && (2.58) \end{aligned}$$

with

$$\mathsf{U}(\mathbf{x}) = \mathsf{S}(\mathbf{x}) C_\infty^* \mathsf{S}(\mathbf{x}) = \begin{pmatrix} -x_h^2 I_2 & x_h \boldsymbol{x}_0 \\ x_h \boldsymbol{x}_0^T & -\boldsymbol{x}_0^T \boldsymbol{x}_0 \end{pmatrix} \tag{2.59}$$

Note that this expression is linear in $\mathbf{y}$ but not in $\mathbf{x}$. Therefore the Jacobian

$$\begin{aligned} \mathsf{V}(\mathbf{x},\mathbf{y}) &= \frac{\partial \mathbf{m}}{\partial \mathbf{x}} \\ &= \begin{pmatrix} x_h y_h I_2 & y_h \boldsymbol{x}_0 - 2x_h \boldsymbol{y}_0 \\ x_h \boldsymbol{y}_0^T - 2y_h \boldsymbol{x}_0^T & \boldsymbol{x}_0^T \boldsymbol{y}_0 \end{pmatrix} \end{aligned} \tag{2.60}$$

is required for error propagation.

The construction of $\mathbf{n}$ is completely analogous by swapping the roles of $\mathbf{x}$ and $\mathbf{y}$ yielding

$$\mathbf{n} = \mathsf{U}(\mathbf{y})\mathbf{x} \tag{2.61}$$

Finally the covariance matrices are obtained by error propagation using the Jacobian

$$\mathsf{J}(\mathbf{x},\mathbf{y}) = \begin{pmatrix} -\mathsf{S}(\mathbf{y}) & \mathsf{S}(\mathbf{x}) \\ \mathsf{V}(\mathbf{x},\mathbf{y}) & \mathsf{U}(\mathbf{x}) \\ \mathsf{U}(\mathbf{y}) & \mathsf{V}(\mathbf{y},\mathbf{x}) \end{pmatrix} \tag{2.62}$$

yielding the covariance matrices

$$\begin{pmatrix} \mathsf{C}_{ll} & \mathsf{C}_{lm} & \mathsf{C}_{ln} \\ \mathsf{C}_{ml} & \mathsf{C}_{mm} & \mathsf{C}_{mn} \\ \mathsf{C}_{nl} & \mathsf{C}_{nm} & \mathsf{C}_{nn} \end{pmatrix} = \mathsf{J}(\mathbf{x},\mathbf{y}) \begin{pmatrix} \mathsf{C}_{xx} & \mathsf{C}_{xy} \\ \mathsf{C}_{yx} & \mathsf{C}_{yy} \end{pmatrix} \mathsf{J}(\mathbf{x},\mathbf{y})^T \tag{2.63}$$

Note that the covariances between the entities are not necessary for the statistical tests presented in section 2.2.2 and therefore often need not to be computed.

Converting from line to point representation: To convert a line segment from line representation to point representation one has to intersect the line $\mathbf{m}$ and $\mathbf{n}$ with $\mathbf{l}$. Therefore the end-points are obtained as (cf. equation (2.38))

$$\mathbf{x} = \mathsf{S}(\mathbf{l})\mathbf{m} = -\mathsf{S}(\mathbf{m})\mathbf{l} \tag{2.64}$$

and

$$\mathbf{y} = -\mathsf{S}(\mathbf{l})\mathbf{n} = \mathsf{S}(\mathbf{n})\mathbf{l} \tag{2.65}$$

Using the Jacobian

$$\mathsf{J}(\mathbf{l},\mathbf{m},\mathbf{n}) = \begin{pmatrix} -\mathsf{S}(\mathbf{m}) & \mathsf{S}(\mathbf{l}) & 0 \\ \mathsf{S}(\mathbf{n}) & 0 & -\mathsf{S}(\mathbf{l}) \end{pmatrix} \tag{2.66}$$

the covariance matrices are obtained as

$$\begin{pmatrix} \mathsf{C}_{xx} & \mathsf{C}_{xy} \\ \mathsf{C}_{yx} & \mathsf{C}_{yy} \end{pmatrix} = \mathsf{J}(\mathbf{l},\mathbf{m},\mathbf{n}) \begin{pmatrix} \mathsf{C}_{ll} & \mathsf{C}_{lm} & \mathsf{C}_{ln} \\ \mathsf{C}_{ml} & \mathsf{C}_{mm} & \mathsf{C}_{mn} \\ \mathsf{C}_{nl} & \mathsf{C}_{nm} & \mathsf{C}_{nn} \end{pmatrix} \mathsf{J}(\mathbf{l},\mathbf{m},\mathbf{n})^T \tag{2.67}$$

Again the covariance between the entities are often not needed for statistical testing and need not be computed.

It has been shown, how line segments can be represented in an uncertain oriented projective framework. Two alternative representations have been presented that can easily be converted into each other and that are both useful for different tasks, as will become clear in section 2.2.

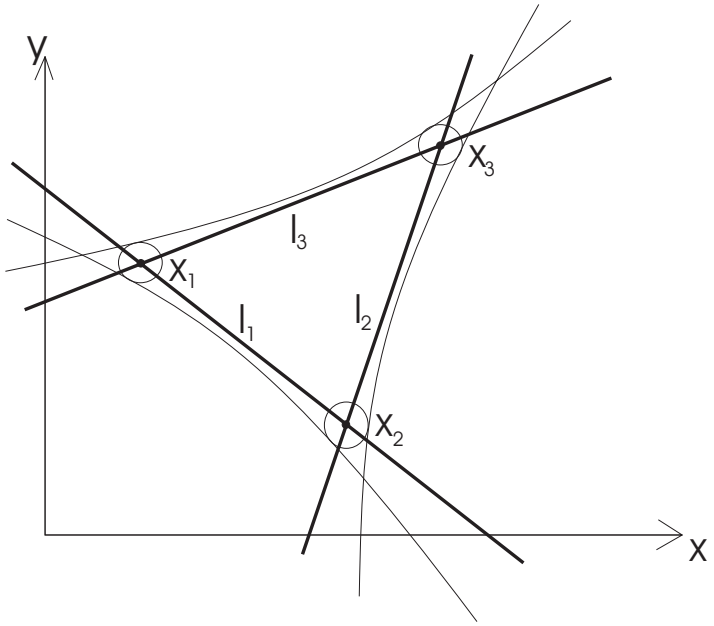

Figure 2.4: An uncertain oriented projective 2d triangle may be represented either by its three end-points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ or by its three delimiting lines $\mathbf{l}_1$, $\mathbf{l}_2$ and $\mathbf{l}_3$. Observe that the confidence region has a different shape in the two different representations.

2d polygons

The third uncertain oriented projective compound entity that will be presented here is the 2d polygon. For instance homogeneous image regions, which are the output of some feature extraction methods, can be modeled as such uncertain oriented projective polygons.

A 2d polygon with N corners can be seen as a collection of N delimiting line segments, where two line segments have one point in common. This yields two natural representations for uncertain polygons, namely using the corner-points itself (see figure 2.4)

$$(\{\mathbf{x}_1, ..., \mathbf{x}_N\}, \{\mathsf{C}_{x_i x_j} | i, j = 1, ..., N\}) \tag{2.68}$$

or using the delimiting lines (see figure 2.4)

$$(\{\mathbf{l}_1, ..., \mathbf{l}_N\}, \{\mathsf{C}_{l_i l_j} | i, j = 1, ..., N\}) \tag{2.69}$$

The conversion from point to line representation is (cf. equation (2.32))

$$\mathbf{l}_i = \mathsf{S}(\mathbf{x}_i)\mathbf{x}_{i+1} = -\mathsf{S}(\mathbf{x}_{i+1})\mathbf{x}_i \tag{2.70}$$

where the indices read modulo N. The covariance matrices are obtained using the Jacobian

$$\mathsf{J}(\mathbf{x}_1, ..., \mathbf{x}_N) = \begin{pmatrix} -\mathsf{S}(\mathbf{x}_2) & \mathsf{S}(\mathbf{x}_1) & & & \\ & \ddots & & & \\ & & & -\mathsf{S}(\mathbf{x}_N) & \mathsf{S}(\mathbf{x}_{N-1}) \\ \mathsf{S}(\mathbf{x}_N) & & & & -\mathsf{S}(\mathbf{x}_1) \end{pmatrix} \tag{2.71}$$

as

$$\left(\mathsf{C}_{l_i l_j}\right)_{i,j=1,...,N} = \mathsf{J}(\mathbf{x}_1, ..., \mathbf{x}_N) \left(\mathsf{C}_{x_i x_j}\right)_{i,j=1,...,N} \mathsf{J}(\mathbf{x}_1, ..., \mathbf{x}_N)^T \tag{2.72}$$

For duality reasons the conversion from line representation into point representation is completely analogous (cf. equation (2.38))

$$\mathbf{x}_i = \mathsf{S}(\mathbf{l}_{i-1})\mathbf{l}_i = -\mathsf{S}(\mathbf{l}_i)\mathbf{l}_{i-1} \tag{2.73}$$

with the covariance being obtained using the Jacobian

$$\mathsf{J}(\mathbf{l}_1, ..., \mathbf{l}_N) = \begin{pmatrix} \mathsf{S}(\mathbf{l}_N) & & & -\mathsf{S}(\mathbf{l}_1) \\ -\mathsf{S}(\mathbf{l}_2) & \mathsf{S}(\mathbf{l}_1) & & \\ & & \ddots & \\ & & -\mathsf{S}(\mathbf{l}_N) & \mathsf{S}(\mathbf{l}_{N-1}) \end{pmatrix} \tag{2.74}$$

as

$$\left(\mathsf{C}_{x_i x_j}\right)_{i,j=1,...,N} = \mathsf{J}(\mathbf{l}_1, ..., \mathbf{l}_N) \left(\mathsf{C}_{l_i l_j}\right)_{i,j=1,...,N} \mathsf{J}(\mathbf{l}_1, ..., \mathbf{l}_N)^T \tag{2.75}$$

Again the covariance matrices between all entities are not always required, which simplifies the computation significantly. Also note that also non-convex polygons can be represented like this, although most of the tests presented in section 2.2 require the polygons to be convex.

This section presented uncertain oriented projective 2d entities. First the two base entities, namely 2d points and 2d lines were discussed, followed by three useful compound entities constructed from those base entities. In the following a completely analogous derivation will be presented for 3d entities.

2.1.3 Uncertain oriented 3d base entities

As before in the 2d case, this section will present the representation of 3d entities in the uncertain oriented projective framework. First, the base entities 3d point, 3d line and 3d plane will be discussed.

3d points

In computer vision and photogrammetry uncertain 3d points are the output of most reconstruction algorithms. Furthermore, there exist many different sensors, such as laser scanners or GPS, that directly yield 3d coordinates as observations.

Such a 3d point is usually given by its Eclidean coordinates (X, Y, Z) together with their variances σ_X^2, σ_Y^2, σ_Z^2 and their covariances σ_{XY}, σ_{XZ}, σ_{YZ}. Hence, the uncertain Euclidean entity is represented as (see figure 2.5)

$$(\boldsymbol{X}, \mathsf{C}_{XX}) = \left(\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \begin{pmatrix} \sigma_X^2 & \sigma_{XY} & \sigma_{XZ} \\ \sigma_{XY} & \sigma_Y^2 & \sigma_{YZ} \\ \sigma_{XZ} & \sigma_{YZ} & \sigma_Z^2 \end{pmatrix} \right) \tag{2.76}$$

The corresponding uncertain oriented projective entity is obtained as (cf. equation (2.26) and [Heuel, 2004, p.106])

$$(\mathbf{X}, \mathsf{C}_{XX}) = \left(\begin{pmatrix} \boldsymbol{X} \\ 1 \end{pmatrix}, \begin{pmatrix} \mathsf{C}_{XX} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} \right) \tag{2.77}$$

Given on the other hand an uncertain oriented projective 3d point

$$(\mathbf{X}, \mathsf{C}_{XX}) = \left(\begin{pmatrix} \boldsymbol{X}_0 \\ X_h \end{pmatrix}, \mathsf{C}_{XX} \right) \tag{2.78}$$

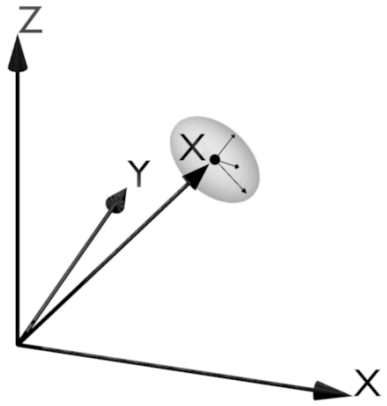

Figure 2.5: An uncertain Euclidean 3d point is represented by its position $\boldsymbol{X}$ and a confidence ellipsoid represented by the covariance matrix C_{XX}. The ellipsoid needs not to be aligned with the coordinate system.

this may be normalized into Euclidean coordinates dividing by $|X_h|$ resulting in the normalized 3d point (cf. equation (2.28) and [Heuel, 2004, p.109f])

$$(\mathbf{X}^{(e)}, \mathsf{C}_{XX}^{(e)}) = \left(\frac{1}{|X_h|} \begin{pmatrix} \boldsymbol{X}_0 \\ X_h \end{pmatrix}, \mathsf{J}_e \mathsf{C}_{XX} \mathsf{J}_e^T \right) \tag{2.79}$$

$$= \left(\begin{pmatrix} \boldsymbol{X}_0^{(e)} \\ \operatorname{sign}(X_h) \end{pmatrix}, \begin{pmatrix} \mathsf{C}_{X_0X_0}^{(e)} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} \right) \tag{2.80}$$

using the Jacobian

$$\mathsf{J}_e = \frac{\partial \mathbf{X}^{(e)}}{\partial \mathbf{X}} = \frac{1}{|X_h|} \begin{pmatrix} \mathsf{I}_3 & -\frac{\boldsymbol{X}_0}{|X_h|} \\ \mathbf{0}^T & 0 \end{pmatrix} \tag{2.81}$$

This is completely analogous to the 2d point and again the orientation is preserved by this operation, so that an Euclidean point is only obtained, if the homogeneous coordinate is positive. Hence, again it is possible to represent every uncertain Euclidean 3d point in the uncertain oriented projective framework.

3d lines

A second very useful entity is the 3d line. In computer vision and photogrammetry it occurs for instance as backprojection of image points (see section 2.1.5). Furthermore, straight lines often occur in man-made environments, so that reconstruction algorithms for such scenes should be able to represent uncertain 3d lines.

An uncertain oriented projective 3d line is obtained by connecting two 3d points. Using

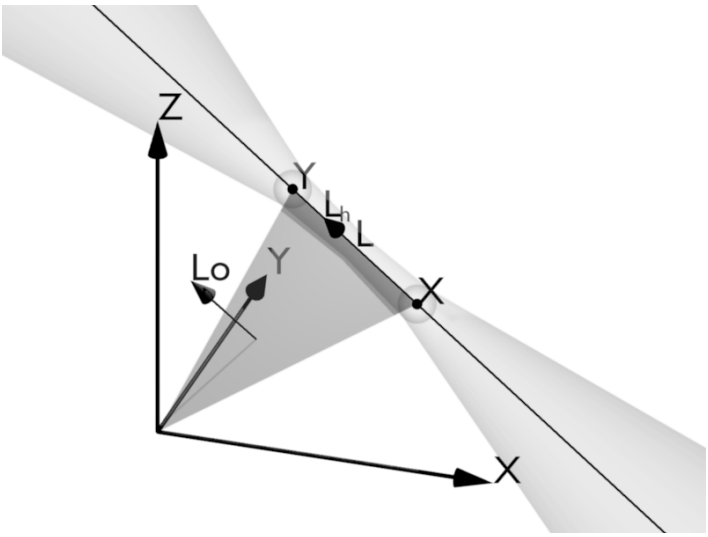

Figure 2.6: An uncertain directed line going from point $\mathbf{X}$ to point $\mathbf{Y}$. The vector $\boldsymbol{L}_h$ indicates the direction of the line and the vector $\boldsymbol{L}_0$ is the normal of the plane going through the line and the origin.

the matrix (cf. [Heuel, 2004, p.52])

$$\mathbf{\Pi}(\mathbf{X}) = \begin{pmatrix} X_4 & 0 & 0 & -X_1 \\ 0 & X_4 & 0 & -X_2 \\ 0 & 0 & X_4 & -X_3 \\ 0 & -X_3 & X_2 & 0 \\ X_3 & 0 & -X_1 & 0 \\ -X_2 & X_1 & 0 & 0 \end{pmatrix} \tag{2.82}$$

the line from point $(\mathbf{X}, \mathsf{C}_{XX})$ to the point $(\mathbf{Y}, \mathsf{C}_{YY})$ is defined as $(\mathbf{L}, \mathsf{C}_{LL})$ with (cf. [Heuel, 2004, p.51] and see figure 2.6)

$$\mathbf{L} = \mathbf{\Pi}(\mathbf{X})\mathbf{Y} = -\mathbf{\Pi}(\mathbf{Y})\mathbf{X} \tag{2.83}$$

and in case of uncorrelated points $\mathbf{X}$ and $\mathbf{Y}$ its covariance matrix is given by

$$\mathsf{C}_{LL} = \mathbf{\Pi}(\mathbf{X})\mathsf{C}_{YY}\mathbf{\Pi}(\mathbf{X})^T + \mathbf{\Pi}(\mathbf{Y})\mathsf{C}_{XX}\mathbf{\Pi}(\mathbf{Y})^T \tag{2.84}$$

Note that changing the order of the two points changes the sign and thus the direction of the connecting line. Not all 6-vectors are constructible from two homogeneous points. This is because the Plücker constraint must be fulfilled, i.e. a 6-vector $\mathbf{L} = (\boldsymbol{L}_h^T, \boldsymbol{L}_0^T)^T$ only represents a line, if and only if the upper 3d vector and the lower 3d vector are perpendicular, i.e.

$$\boldsymbol{L}_h^T \boldsymbol{L}_0 = 0 \tag{2.85}$$

An Euclidean interpretation of the uncertain oriented projective 3d line

$$(\mathbf{L}, \mathsf{C}_{LL}) = \left(\begin{pmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_0 \end{pmatrix}, \mathsf{C}_{LL} \right) \tag{2.86}$$

is possible by dividing by the norm of its homogeneous part $|\boldsymbol{L}_h|$ resulting in the normalized 3d line (cf. [Heuel, 2004, p.109f])

$$(\mathbf{L}^{(e)}, \mathsf{C}_{LL}^{(e)}) = \left(\frac{1}{|\boldsymbol{L}_h|} \begin{pmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_0 \end{pmatrix}, J_e \mathsf{C}_{LL} J_e^T \right) \tag{2.87}$$

$$= \left(\begin{pmatrix} \boldsymbol{L}_h^{(e)} \\ \boldsymbol{L}_0^{(e)} \end{pmatrix}, \mathsf{C}_{LL}^{(e)} \right) \tag{2.88}$$

using the Jacobian

$$J_e = \frac{\partial \mathbf{L}^{(e)}}{\partial \mathbf{L}} = \frac{1}{|\boldsymbol{L}_h|} \begin{pmatrix} I_3 - \frac{\boldsymbol{L}_h \boldsymbol{L}_h^T}{|\boldsymbol{L}_h|^2} & 0 \\ \frac{-\boldsymbol{L}_h \boldsymbol{L}_0^T}{|\boldsymbol{L}_h|^2} & I_3 \end{pmatrix} \tag{2.89}$$

This yields the oriented line direction $\boldsymbol{L}_h^{(e)}$, which can be interpreted using spherical coordinates

$$\boldsymbol{L}_h^{(e)} = \begin{pmatrix} \cos\theta \sin\phi \\ \sin\theta \sin\phi \\ \cos\phi \end{pmatrix} \tag{2.90}$$

with θ being the oriented angle in the xy-plane and ϕ being the oriented angle out of this plane, as well as the point on the line being closest to the origin (cf. [Förstner and Wrobel, 2004, p.120])

$$\boldsymbol{Z} = \boldsymbol{L}_h^{(e)} \times \boldsymbol{L}_0^{(e)} \tag{2.91}$$

It has been seen, how an uncertain oriented projective 3d line can be represented and constructed from two uncertain oriented projective 3d points. It has also been shown, how the representation can be interpreted in an Euclidean setting. Next, the uncertain oriented projective 3d plane will be presented that results from joining a 3d line with a 3d point.

3d planes

The third important 3d base entity is the 3d plane. Like the 3d line it occurs in many man-made environments and is also encountered in computer vision and photogrammetry as the backprojection of image lines (see section 2.1.5).

An uncertain oriented projective 3d plane is obtained by connecting a 3d line with a 3d point. Using the matrix (cf. [Heuel, 2004, p.52])

$$\overline{\mathbf{\Pi}}(\mathbf{X}) = D_6 \mathbf{\Pi}(\mathbf{X}) = \begin{pmatrix} 0 & -X_3 & X_2 & 0 \\ X_3 & 0 & -X_1 & 0 \\ -X_2 & X_1 & 0 & 0 \\ X_4 & 0 & 0 & -X_1 \\ 0 & X_4 & 0 & -X_2 \\ 0 & 0 & X_4 & -X_3 \end{pmatrix} \tag{2.92}$$

with the dualizing matrix (cf. [Heuel, 2004, p.44])

$$D_6 = \begin{pmatrix} 0 & I_3 \\ I_3 & 0 \end{pmatrix} \tag{2.93}$$

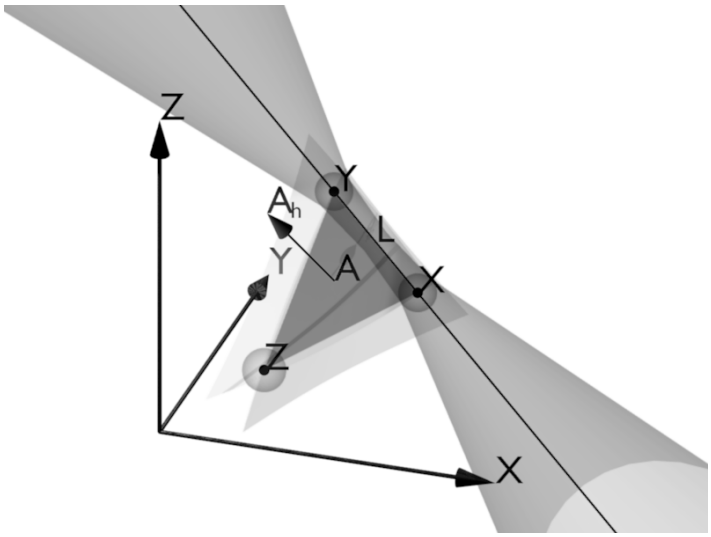

Figure 2.7: An uncertaint oriented 3d plane $\mathbf{A}$ constructed from an uncertain directed line $\mathbf{L}$ and an uncertain point $\mathbf{Z}$. The vector $\boldsymbol{A}_h$ is the normal of the plane defining its orientation.

and the matrix (cf. [Heuel, 2004, p.53])

$$\overline{\mathbf{\Gamma}}(\mathbf{L}) = \begin{pmatrix} 0 & L_3 & -L_2 & -L_4 \\ -L_3 & 0 & L_1 & -L_5 \\ L_2 & -L_1 & 0 & -L_6 \\ L_4 & L_5 & L_6 & 0 \end{pmatrix} \tag{2.94}$$

the plane connecting the line $(\mathbf{L}, \mathsf{C}_{LL})$ with the point $(\mathbf{X}, \mathsf{C}_{XX})$ is defined as $(\mathbf{A}, \mathsf{C}_{AA})$ with (cf. [Heuel, 2004, p.53] and see figure 2.7)

$$\mathbf{A} = \overline{\mathbf{\Pi}}(\mathbf{X})^T \mathbf{L} = \overline{\mathbf{\Gamma}}(\mathbf{L})^T \mathbf{X} \tag{2.95}$$

and

$$\mathsf{C}_{AA} = \overline{\mathbf{\Pi}}(\mathbf{X})^T \mathsf{C}_{LL} \overline{\mathbf{\Pi}}(\mathbf{X}) + \overline{\mathbf{\Gamma}}(\mathbf{L})^T \mathsf{C}_{XX} \overline{\mathbf{\Gamma}}(\mathbf{L})) \tag{2.96}$$

if the point $\mathbf{X}$ is uncorrelated with the line $\mathbf{L}$. Note that the orientation of the plane depends on the orientation of the line and the orientation of the point, which will become clearer in the Euclidean interpretation.

To obtain this Euclidean interpretation of a given uncertain oriented 3d plane

$$(\mathbf{A}, \mathsf{C}_{AA}) = \left(\begin{pmatrix} \boldsymbol{A}_h \\ A_0 \end{pmatrix}, \mathsf{C}_{AA} \right) \tag{2.97}$$

it is divided by the norm of its homogeneous part $|\boldsymbol{A}_h|$ resulting in the normalized plane (cf. [Heuel, 2004, p.109f])

$$(\mathbf{A}^{(e)}, \mathsf{C}_{AA}^{(e)}) = \left(\frac{1}{|\boldsymbol{A}_h|} \begin{pmatrix} \boldsymbol{A}_h \\ A_0 \end{pmatrix}, J_e \mathsf{C}_{AA} J_e^T \right) \tag{2.98}$$

$$= \left(\begin{pmatrix} \boldsymbol{A}_h^{(e)} \\ A_0^{(e)} \end{pmatrix}, \mathsf{C}_{AA}^{(e)} \right) \tag{2.99}$$

using the Jacobian

$$J_e = \frac{\partial \mathbf{A}^{(e)}}{\partial \mathbf{A}} = \frac{1}{|\boldsymbol{A}_h|} \begin{pmatrix} I_3 - \frac{\boldsymbol{A}_h \boldsymbol{A}_h^T}{|\boldsymbol{A}_h|^2} & \mathbf{0} \\ \frac{-A_0 \boldsymbol{A}_h^T}{|\boldsymbol{A}_h|^2} & 1 \end{pmatrix} \tag{2.100}$$

This yields the directed normal $\boldsymbol{A}_h^{(e)}$ of the plane, which can be interpreted in terms of spherical coordinates

$$\boldsymbol{A}_h^{(e)} = \begin{pmatrix} \cos\theta \sin\phi \\ \sin\theta \sin\phi \\ \cos\phi \end{pmatrix} \tag{2.101}$$

and the directed distance to the origin $A_0^{(e)}$, where the sign of $A_0^{(e)}$ is positive, if the plane's normal points toward the origin and negative if it points away from the origin.

The orientation of the plane, i.e. the direction of its normal vector $\boldsymbol{A}_h$, defines a notion of above and below the plane with the convention that above is in the direction the normal vector points, which is consistent with the 2d case as will seen in detail in section 2.2.1.

Having defined the uncertain oriented 3d plane some more useful construction rules can be defined. For instance given three uncertain 3d points $(\mathbf{X}, \mathsf{C}_{XX})$, $(\mathbf{Y}, \mathsf{C}_{YY})$ and $(\mathbf{Z}, \mathsf{C}_{ZZ})$, the plane going through them is obtained by first connecting the first two points into a line (cf. equation (2.83)) and then this line with the third point. Being more precise, the plane is obtained as $(\mathbf{A}, \mathsf{C}_{AA})$ with (cf. equation (2.95) and [Heuel, 2004, p.55])

$$\mathbf{A} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X})\mathbf{Y})^T\mathbf{Z} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{Y})\mathbf{Z})^T\mathbf{X} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{Z})\mathbf{X})^T\mathbf{Y} \tag{2.102}$$

and in case of uncorrelated points $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$

$$\begin{aligned} \mathsf{C}_{AA} \;=\; & \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X})\mathbf{Y})^T \mathsf{C}_{ZZ} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X})\mathbf{Y} \\ & + \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{Y})\mathbf{Z})^T \mathsf{C}_{XX} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{Y})\mathbf{Z} \\ & + \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{Z})\mathbf{X})^T \mathsf{C}_{YY} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{Z})\mathbf{X} \end{aligned} \tag{2.103}$$

The orientation of the resulting plane is such that in case all three points have positive orientation, the direction from $\mathbf{X}$ to $\mathbf{Y}$, the direction from $\mathbf{X}$ to $\mathbf{Z}$ and the normal of the plane form a right handed coordinate system.

For duality reasons the intersection of three uncertain oriented projective 3d planes $(\mathbf{A}, \mathsf{C}_{AA})$, $(\mathbf{B}, \mathsf{C}_{BB})$ and $(\mathbf{C}, \mathsf{C}_{CC})$ can be defined analogously to (2.102) yielding the oriented uncertain projective 3d point $(\mathbf{X}, \mathsf{C}_{XX})$ with (cf. [Heuel, 2004, p.55])

$$\mathbf{X} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{A})\mathbf{B})^T\mathbf{C} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B})\mathbf{C})^T\mathbf{A} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{C})\mathbf{A})^T\mathbf{B} \tag{2.104}$$

and

$$\begin{aligned} \mathsf{C}_{XX} \;=\; & \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{A})\mathbf{B})^T \mathsf{C}_{CC} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{A})\mathbf{B} \\ & + \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B})\mathbf{C})^T \mathsf{C}_{AA} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B})\mathbf{C} \\ & + \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{C})\mathbf{A})^T \mathsf{C}_{BB} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{C})\mathbf{A} \end{aligned} \tag{2.105}$$

if the three planes are assumed to be uncorrelated. The orientation of the resulting point is positive, if and only if the normals of the planes $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ form a left-handed coordinate system.

Dualizing the definition of the uncertain oriented projective 3d plane (cf. equation (2.95)) yields the intersection of an oriented projective 3d line $(\mathbf{L}, \mathsf{C}_{LL})$ and an oriented projective 3d

plane $(\mathbf{A}, \mathsf{C}_{AA})$ being the uncertain oriented projective 3d point $(\mathbf{X}, \mathsf{C}_{XX})$ with (cf. [Heuel, 2004, p.55])

$$\mathbf{X} = \mathbf{\Pi}(\mathbf{A})^T \mathbf{L} = \mathbf{\Gamma}(\mathbf{L})^T \mathbf{A} \tag{2.106}$$

and in case of uncorrelated $\mathbf{L}$ and $\mathbf{A}$

$$\mathsf{C}_{XX} = \mathbf{\Pi}(\mathbf{A})^T \mathsf{C}_{LL} \mathbf{\Pi}(\mathbf{A}) + \mathbf{\Gamma}(\mathbf{L})^T \mathsf{C}_{AA} \mathbf{\Gamma}(\mathbf{L}) \tag{2.107}$$

using the dualized matrix (cf. [Heuel, 2004, p.53])

$$\mathbf{\Gamma}(\mathbf{L}) = \overline{\mathbf{\Gamma}}(D_6 \mathbf{L}) = \begin{pmatrix} 0 & L_6 & -L_5 & -L_1 \\ -L_6 & 0 & L_4 & -L_2 \\ L_5 & -L_4 & 0 & -L_3 \\ L_1 & L_2 & L_3 & 0 \end{pmatrix} \tag{2.108}$$

The orientation of the resulting point is positive, if and only if the line pierces the plane from left as will be discussed in detail in section 2.2.1.

Finally also the definition of the uncertain oriented projective 3d line (cf. equation (2.83)) can be dualized yielding the intersection of the two oriented projective 3d planes $(\mathbf{A}, \mathsf{C}_{AA})$ and $(\mathbf{B}, \mathsf{C}_{BB})$ being the line $(\mathbf{L}, \mathsf{C}_{LL})$ with (cf. [Heuel, 2004, p.55])

$$\mathbf{L} = \overline{\mathbf{\Pi}}(\mathbf{A})\mathbf{B} = -\overline{\mathbf{\Pi}}(\mathbf{B})\mathbf{A} \tag{2.109}$$

and

$$\mathsf{C}_{LL} = \overline{\mathbf{\Pi}}(\mathbf{A}) \mathsf{C}_{BB} \overline{\mathbf{\Pi}}(\mathbf{A})^T + \overline{\mathbf{\Pi}}(\mathbf{B}) \mathsf{C}_{AA} \overline{\mathbf{\Pi}}(\mathbf{B})^T \tag{2.110}$$

if the two planes are uncorrelated. The orientation of the resulting line is such that the normal vector of the plane $\mathbf{A}$, the normal vector of the plane $\mathbf{B}$ and the direction vector of the line form a right handed coordinate system.

Up to now the three uncertain oriented projective 3d base entities point, line and plane have been presented. It has been shown, how they could be constructed and how they could be interpreted in an Euclidean setting. As in the 2d case, those entities can be combined into useful compound entities, which will be presented in the following section.

2.1.4 Uncertain oriented 3d compound entities

Now the uncertain oriented projective 3d entities presented in the previous section will be combined into uncertain oriented projective compound entities. It will be shown, how 3d facets, 3d line segments and 3d polygons can be represented in the uncertain oriented projective framework.

3d facets

Analogous to the 2d edgels facets, i.e. 3d points with an associated normal direction, may be defined in 3d space. Facets are for instance required for visualization purposes in computer graphics and are usually obtained from laser scanner data from the known neighborhood structure within the 3d point cloud.

Those 3d facets are constructed from an uncertain oriented projective 3d point $(\mathbf{X}, \mathsf{C}_{XX})$ together with a local surface normal

$$(\mathbf{T}, \mathsf{C}_{TT}) = \left(\begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix}, \mathsf{C}_{TT} \right) \tag{2.111}$$

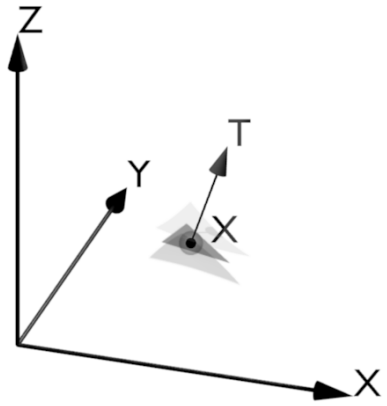

Figure 2.8: An uncertain oriented projective 3d facet located at the point $\mathbf{X}$ having the normal $\mathbf{T}$.

being an uncertain oriented projective 2d point representing the local surface normal. The resulting 3d facet is then defined as the compound oriented projective entity (see figure 2.8)

$$(\{\mathbf{X}, \mathbf{T}\}, \{\mathsf{C}_{XX}, \mathsf{C}_{TT}, \mathsf{C}_{XT}\}) \tag{2.112}$$

The point at infinity corresponding to the normal direction vector is computable in complete analogy to the 2d edgel case using the matrix (cf. 2.43)

$$\tilde{\mathsf{C}}_\infty = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.113}$$

as

$$\mathbf{X}_\infty = \tilde{\mathsf{C}}_\infty \mathbf{T} \tag{2.114}$$

having the covariance matrix

$$\mathsf{C}_{X_\infty X_\infty} = \tilde{\mathsf{C}}_\infty \mathsf{C}_{TT} \tilde{\mathsf{C}}_\infty^T \tag{2.115}$$

The plane $\mathbf{A}$ going through the facet has the normal vector $\mathbf{T}$ and goes through $\mathbf{X}$, hence

$$\mathbf{A}^T \mathbf{X} = \begin{pmatrix} \boldsymbol{T} \\ A_0 \end{pmatrix}^T \begin{pmatrix} \boldsymbol{X}_0 \\ X_h \end{pmatrix} = 0 \tag{2.116}$$

Solving this equation for the unknown distance A_0 yields

$$A_0 = -\frac{\boldsymbol{T}^T \boldsymbol{X}_0}{X_h} \tag{2.117}$$

Multiplying with the common factor X_h yields the plane

$$\mathbf{A} = \begin{pmatrix} X_h \boldsymbol{T} \\ -\boldsymbol{T}^T \boldsymbol{X}_0 \end{pmatrix} = \mathsf{U}(\mathbf{X})\mathbf{T} = \mathsf{V}(\mathbf{T})\mathbf{X} \tag{2.118}$$

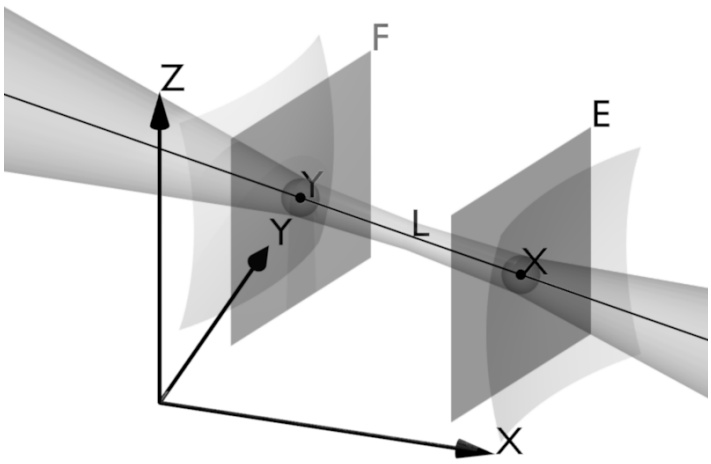

Figure 2.9: An uncertain oriented 3d line segment can either be represented by its two end-points **X** and **Y** or by the line **L** and the two delimiting planes **E** and **F**. Note that like the 2d case the shape of the confidence region depends on the representation.

with the Jacobians

$$U(\mathbf{X}) = \begin{pmatrix} X_h I_3 \\ -\boldsymbol{X}_0 \end{pmatrix} \tag{2.119}$$

and

$$V(\mathbf{T}) = \begin{pmatrix} 0_{3\times 3} & \boldsymbol{T} \\ -\boldsymbol{T}^T & 0 \end{pmatrix} \tag{2.120}$$

Note that X_h might be negative, so that a point with negative orientation results in a plane with a normal vector pointing in the opposite direction of the normal vector of the facet. The covariance matrix of the resulting plane is finally obtained under the assumption that the facets normal is uncorrelated with its position, as

$$\mathsf{C}_{AA} = U(\mathbf{X})\mathsf{C}_{TT}U(\mathbf{X})^T + V(\mathbf{T})\mathsf{C}_{XX}V(\mathbf{T})^T \tag{2.121}$$

The additional information contained in the surface normal can often be used to great advantage. As in the case of 2d edgels, grouping is significantly eased, if such normal direction information is available in addition to the position of a 3d point. Furthermore, realistic visualization of 3d point clouds is only possible, if the normal directions are known for each 3d point, as the reflectance properties of the surface strongly depend on it.

3d line segments

Like in the 2d case, also 3d line segments may be represented in the uncertain oriented projective framework as compound entities. Again two different representations for 3d line segments will be required in section 2.2 and will be presented in the following.

Line segments in point representation: As in the 2d case a 3d line segment is usually constructed from its two end-points. Again it does not extend from the end-points in two directions but may cross the plane at infinity and have therefore infinite length in an Euclidean interpretation, too. Constructing the compound entity from the ordered pair of

uncertain oriented projective 3d end-points $(\mathbf{X}, \mathsf{C}_{XX})$ and $(\mathbf{Y}, \mathsf{C}_{YY})$, one obtains (see figure 2.9)

$$(\{\mathbf{X}, \mathbf{Y}\}, \{\mathsf{C}_{XX}, \mathsf{C}_{YY}, \mathsf{C}_{XY}\}) \tag{2.122}$$

All points are considered on the line segment that starting from point $\mathbf{X}$ lie along the directed line $\mathbf{\Pi}(\mathbf{X})\mathbf{Y}$ until the point $\mathbf{Y}$ is reached. As in the 2d case the plane at infinity is crossed, if and only if the orientation of $\mathbf{X}$ and $\mathbf{Y}$ differs.

Line segments in line-plane representation: The second useful representation for a 3d line segment is the compound entity comprising of the line $(\mathbf{L}, \mathbf{C}_{LL})$ connecting its two end-points and the two planes $(\mathbf{E}, \mathsf{C}_{EE})$ and $(\mathbf{F}, \mathsf{C}_{FF})$ going perpendicular to the line through the end-points (see figure 2.9)

$$(\{\mathbf{L}, \mathbf{E}, \mathbf{F}\}, \{\mathsf{C}_{LL}, \mathsf{C}_{EE}, \mathsf{C}_{FF}, \mathsf{C}_{LE}, \mathsf{C}_{LF}, \mathsf{C}_{EF}\}) \tag{2.123}$$

Because of the perpendicularity constraint, not all such entities represent 3d line segments. For defining orthogonality between lines and planes, one can use the $3 \times 3 \times 3$-tensor (cf. [Hartley and Zisserman, 2000, p.546])

$$\epsilon_{rst} = \begin{cases} 0 & \text{unless } r, s \text{ and } t \text{ are distinct} \\ +1 & \text{if } rst \text{ is an even permutation of } 123 \\ -1 & \text{if } rst \text{ is an odd permutation of } 123 \end{cases} \tag{2.124}$$

and augment it with zeros to obtain a $3 \times 6 \times 4$-tensor

$$\tilde{\epsilon}_{rst} = \begin{cases} \epsilon_{rst} & \text{if } s \leq 3 \text{ and } t \leq 3 \\ 0 & \text{otherwise} \end{cases} \tag{2.125}$$

The two perpendicularity constraints can now be stated as

$$\tilde{\epsilon}_{ijk} L_i E_k = 0 \tag{2.126}$$

and

$$\tilde{\epsilon}_{ijk} L_i F_k = 0 \tag{2.127}$$

Note that from each of those constraint only two are linear independent and, as orthogonality is an Euclidean property, none of the end-points are allowed to lie at infinity in this representation. Also note that the two constraints are also expressible as cross-products, but are given in tensor notation to ease the following derivations.

Converting from point to line-plane representation: To obtain a 3d line segment in line-plane representation, one first constructs the connecting line as shown above (cf. equation (2.83))

$$\mathbf{L} = \mathbf{\Pi}(\mathbf{X})\mathbf{Y} = -\mathbf{\Pi}(\mathbf{Y})\mathbf{X} \tag{2.128}$$

From the perpendicularity constraint (cf. 2.126) in tensor notation follows immediately that the three ideal points

$$X'_k = \tilde{\epsilon}_{1jk} L_j \tag{2.129}$$
$$X''_k = \tilde{\epsilon}_{2jk} L_j \tag{2.130}$$
$$X'''_k = \tilde{\epsilon}_{3jk} L_j \tag{2.131}$$

lie on $\mathbf{E}$. Because the end-points are assumed to be not at infinity they are distinct from $\mathbf{X}$. Therefore the plane $\pm\mathbf{E}$ can be constructed up to its orientation by connecting $\mathbf{X}$ with two

of those three points. Selecting without loss of generality the first two points, it is given by (cf. 2.102)

$$\begin{aligned} \pm\mathbf{E} &= \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}')\mathbf{X}'')^T\mathbf{X} && (2.132)\\ &= d_3 \begin{pmatrix} d_1X_h \\ d_2X_h \\ d_3X_h \\ -d_1X_1 - d_2X_2 - d_3X_3 \end{pmatrix} && (2.133) \end{aligned}$$

using the 2×2 - determinant

$$d_i = \begin{vmatrix} X_i & Y_i \\ X_h & Y_h \end{vmatrix} \tag{2.134}$$

The scale factor d_3, which would be d_2 if the first and third point had been selected and d_1 if the second and third point had been selected, can now be omitted. As a byproduct, the scale ambiguity resulting from the selected points at infinity is resolved, so that the oriented plane can now in complete analogy to the 2d case be defined as

$$\mathbf{E} = \begin{pmatrix} d_1X_h \\ d_2X_h \\ d_3X_h \\ -d_1X_1 - d_2X_2 - d_3X_3 \end{pmatrix} = \mathsf{U}(\mathbf{X})\mathbf{Y} \tag{2.135}$$

with

$$\mathsf{U}(\mathbf{X}) = \begin{pmatrix} -X_h^2 I_3 & X_h\boldsymbol{X}_0 \\ X_h\boldsymbol{X}_0^T & -\boldsymbol{X}_0^T\boldsymbol{X}_0 \end{pmatrix} \tag{2.136}$$

Observe that this matrix has the identical form as in the 2d case (cf. equation (2.59)). Again this expression is linear in $\mathbf{Y}$, but not in $\mathbf{X}$. Therefore the Jacobian

$$\begin{aligned} \mathsf{V}(\mathbf{X},\mathbf{Y}) &= \frac{\partial \mathbf{E}}{\partial \mathbf{X}} && (2.137)\\ &= \begin{pmatrix} X_hY_hI_3 & Y_h\boldsymbol{X}_0 - 2X_h\boldsymbol{Y}_0 \\ X_h\boldsymbol{Y}_0^T - 2Y_h\boldsymbol{X}_0^T & \boldsymbol{X}_0^T\boldsymbol{Y}_0 \end{pmatrix} \end{aligned}$$

is required for error propagation.

The construction of $\mathbf{F}$ is completely analogous by swapping the roles of $\mathbf{X}$ and $\mathbf{Y}$ yielding

$$\mathbf{F} = \mathsf{U}(\mathbf{Y})\mathbf{X} \tag{2.138}$$

As in the 2d case the covariance matrices are obtained by error propagation using the Jacobian

$$\mathsf{J}(\mathbf{X},\mathbf{Y}) = \begin{pmatrix} -\mathbf{\Pi}(\mathbf{Y}) & \mathbf{\Pi}(\mathbf{X}) \\ \mathsf{V}(\mathbf{X},\mathbf{Y}) & \mathsf{U}(\mathbf{X}) \\ \mathsf{U}(\mathbf{Y}) & \mathsf{V}(\mathbf{Y},\mathbf{X}) \end{pmatrix} \tag{2.139}$$

yielding

$$\begin{pmatrix} \mathsf{C}_{LL} & \mathsf{C}_{LE} & \mathsf{C}_{LF} \\ \mathsf{C}_{EL} & \mathsf{C}_{EE} & \mathsf{C}_{EF} \\ \mathsf{C}_{FL} & \mathsf{C}_{FE} & \mathsf{C}_{FF} \end{pmatrix} = \mathsf{J}(\mathbf{X},\mathbf{Y}) \begin{pmatrix} \mathsf{C}_{XX} & \mathsf{C}_{XY} \\ \mathsf{C}_{YX} & \mathsf{C}_{YY} \end{pmatrix} \mathsf{J}(\mathbf{X},\mathbf{Y})^T \tag{2.140}$$

Note that often the covariances between the entities are not required for statistical testing and can be omitted from the computation.

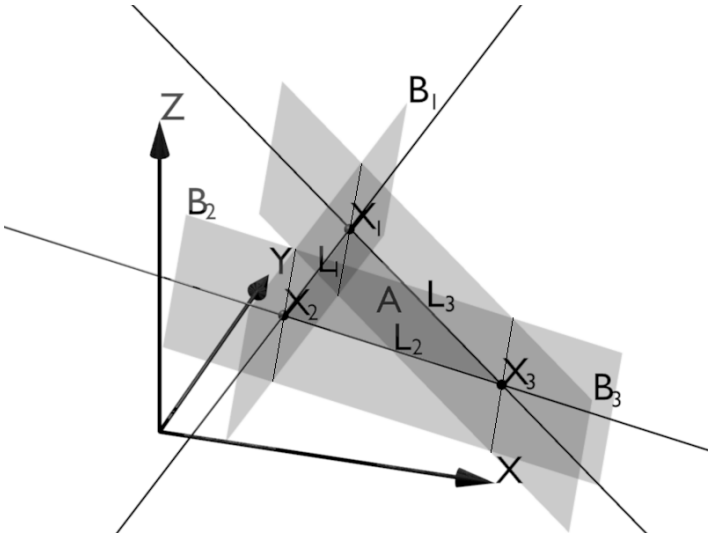

Figure 2.10: An uncertain oriented 3d triangle can be represented by its three corner-points $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$, by its three delimiting lines $\mathbf{L}_1$, $\mathbf{L}_2$ and $\mathbf{L}_3$ or by the plane $\mathbf{A}$ and the three delimiting planes $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$.

Converting from line-plane to point representation: To convert a line segment from line-plane representation into point representation, one has to intersect the two planes $\mathbf{E}$ and $\mathbf{F}$ with $\mathbf{L}$. The end-points are therefore given by (cf. equation (2.106))

$$\mathbf{X} = \mathbf{\Pi}(\mathbf{E})^T\mathbf{L} = -\mathbf{\Pi}(\mathbf{L})^T\mathbf{E} \tag{2.141}$$

and

$$\mathbf{Y} = -\mathbf{\Pi}(\mathbf{F})^T\mathbf{L} = \mathbf{\Pi}(\mathbf{L})^T\mathbf{F} \tag{2.142}$$

Using the Jacobian

$$\mathsf{J}(\mathbf{L},\mathbf{E},\mathbf{F}) = \begin{pmatrix} -\mathbf{\Pi}(\mathbf{E}) & \mathbf{\Pi}(\mathbf{L}) & 0 \\ \mathbf{\Pi}(\mathbf{F}) & 0 & -\mathbf{\Pi}(\mathbf{L}) \end{pmatrix} \tag{2.143}$$

the covariance matrices are obtained as

$$\begin{pmatrix} \mathsf{C}_{XX} & \mathsf{C}_{XY} \\ \mathsf{C}_{YX} & \mathsf{C}_{YY} \end{pmatrix} = \mathsf{J}(\mathbf{L},\mathbf{E},\mathbf{F}) \begin{pmatrix} \mathsf{C}_{LL} & \mathsf{C}_{LE} & \mathsf{C}_{LF} \\ \mathsf{C}_{EL} & \mathsf{C}_{EE} & \mathsf{C}_{EF} \\ \mathsf{C}_{FL} & \mathsf{C}_{FE} & \mathsf{C}_{FF} \end{pmatrix} \mathsf{J}(\mathbf{L},\mathbf{E},\mathbf{F})^T \tag{2.144}$$

Again the covariances between the entities are not always required and need not be computed.

It has been shown, how 3d line segments can be represented as uncertain oriented projective compound entites. Like in the 2d case, two different representations have been presented and it was shown, how to convert from one representation into the other and vice versa, which will be required for the geometric reasoning presented in section 2.2.

3d polygons

A third very useful compound entity representable in the uncertain oriented projective framework is the 3d polygon. A 3d polygon is a planar patch in 3d space that is delimited by three

or more straight line segments. As in the 2d case, those space polygons need not be convex, although some of the tests presented in section 2.2 require such.

In the following three representations, namely one using points, one using planes and one using lines are presented. It will be shown, how to convert 3d polygons from point to plane representation and vice versa as well as from point to line representation and vice versa. The conversion from plane to line representation are therefore possible via the point representation and will not be discussed.

Representation using points: One can represent a 3d polygon with $N \geq 3$ corners simply using the corner points itself (see figure 2.10)

$$(\{\mathbf{X}_1, ..., \mathbf{X}_N\}, \{\mathsf{C}_{X_i X_j} | i, j = 1, ..., N\}) \tag{2.145}$$

If $N > 3$, then not all such compound entities represent actual polygons, because the points are required to be all coplanar. This condition is formulated algebraically by stating that the determinants are

$$\left| \begin{array}{cccc} \mathbf{X}_i & \mathbf{X}_j & \mathbf{X}_k & \mathbf{X}_l \end{array} \right| = 0 \qquad \forall i, j, k, l \in \{1, ..., N\} \tag{2.146}$$

for all subsets of corner points.

Representation using planes: Another representation for 3d polygons is based on the plane $\mathbf{A}$, in which the polygon lies together with the ordered list of perpendicular delimiting planes $\mathbf{B}_1, ..., \mathbf{B}_N$ yielding the entity (see figure 2.10))

$$(\{\mathbf{A}, \mathbf{B}_1, ..., \mathbf{B}_N\}, \{\mathsf{C}_{AA}, \mathsf{C}_{B_i B_j}, \mathsf{C}_{AB_i} | i, j = 1, ..., N\}) \tag{2.147}$$

Orthogonality can be defined between planes similar to the 2d case using the degenerate quadric

$$\tilde{C}^*_\infty = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.148}$$

by stating that a compound entity represents a 3d polygon if and only if the constraints

$$\mathbf{A}^T \tilde{C}^*_\infty \mathbf{B}_i = 0 \qquad \forall i \in \{1, ..., N\} \tag{2.149}$$

hold. Again, as orthogonality is an Euclidean property, no corner point is allowed to lie at infinity in this representation.

Representation using lines: Another third useful representation for 3d polygons uses the delimiting lines instead of the points. The compound entity is then (see figure 2.10)

$$(\{\mathbf{L}_1, ..., \mathbf{L}_n\}, \{\mathsf{C}_{L_i L_j} | i, j = 1, ..., N\}) \tag{2.150}$$

Not every such entity represents a 3d polygon, since the lines have to be all co-planar. Algebraically this condition is formulated by requiring that all inner products are

$$\mathbf{L}_i^T D_6 \mathbf{L}_j = 0 \qquad \forall i, j \in \{1, ..., N\} \tag{2.151}$$

Converting from point to plane representation: To convert a 3d polygon from point to plane representation, one first constructs the plane going through all corner points.

Without loss of generality choosing the corner points $\mathbf{X}_r$, $\mathbf{X}_s$ and $\mathbf{X}_t$ with $r < s < t$ it is given by (cf. equation (2.102))

$$\mathbf{A} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_r)\mathbf{X}_s)^T\mathbf{X}_t \tag{2.152}$$

This choice is arbitrary for points in general position to calculate the expectation value of the uncertain plane $\mathbf{A}$. As will be discussed below, the second moments of the distribution of $\mathbf{A}$ depend on this choice, so that the conversion is not uniquely possible for polygons other than triangles.

From the perpendicularity constraint (cf. equation (2.149)) follows that the ideal point

$$\mathbf{X}' = \tilde{C}^*_\infty\mathbf{A} \tag{2.153}$$

lies on all planes $\mathbf{B}_i$. Because the corner points are assumed not to be at infinity, it is distinct from each of the corner points. Therefore the delimiting planes are given by connecting every pair of neighboring corner points with $\mathbf{X}'$ (cf. equation (2.102)) yielding

$$\mathbf{B}_i = -\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_{i+1})^T\mathbf{X}' \tag{2.154}$$

where the indices read modulo N. Note that the orientations depend on the orientations and the order of the corner points.

As pointed out above, no unique conversion from point to plane representation is possible for $N > 3$. This is, because the description of the uncertainty of the corner points is much richer and may not be captured by the second moments of a single plane. However, choosing three corner points $\mathbf{X}_r$, $\mathbf{X}_s$ and $\mathbf{X}_t$ defining the polygon plane allows to uniquely compute the second moments of the plane representation. The covariance matrix of this plane is then given by (cf. equation (2.102))

$$\mathsf{C}_{AA} = J \begin{pmatrix} \mathsf{C}_{X_rX_r} & \mathsf{C}_{X_rX_s} & \mathsf{C}_{X_rX_t} \\ \mathsf{C}_{X_sX_r} & \mathsf{C}_{X_sX_s} & \mathsf{C}_{X_sX_t} \\ \mathsf{C}_{X_tX_r} & \mathsf{C}_{X_tX_s} & \mathsf{C}_{X_tX_t} \end{pmatrix} J^T \tag{2.155}$$

using the Jacobian

$$J(\mathbf{X}_r, \mathbf{X}_s, \mathbf{X}_t) = \begin{pmatrix} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_s)\mathbf{X}_t) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_r)\mathbf{X}_t) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_r)\mathbf{X}_s) \end{pmatrix}^T \tag{2.156}$$

Uniqueness could artificially be achieved by introducing some criterion on choosing those three points. If no such criterion is available from the application, one might minimize some measure like the trace or the determinant of the resulting covariance matrix C_{AA}.

For the computation of the covariance matrices of the planes $\mathbf{B}_i$ one has to distinguish three cases

1. Both points $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$ are identical with two of the points $\mathbf{X}_r$, $\mathbf{X}_s$ and $\mathbf{X}_t$.
2. Only one of the points $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$ is identical with one of the points $\mathbf{X}_r$, $\mathbf{X}_s$ and $\mathbf{X}_t$.
3. None of the points $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$ is identical with any of the points $\mathbf{X}_r$, $\mathbf{X}_s$ and $\mathbf{X}_t$.

Observe that in the case of triangles only the first case can occur, and that the third case occurs only with polygons having five of more corner points. Also note, the exchanging $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$ or permuting $\mathbf{X}_r$, $\mathbf{X}_s$ and $\mathbf{X}_t$ only changes the sign of $\mathbf{B}_i$ (cf. equation (2.154))

and therefore does not affect its covariance matrix. Thus the ordering of points will be fixed without loss of generality in the following analysis.

Case 1: Now starting with the first case and assuming without loss of generality that $\mathbf{X}_i = \mathbf{X}_r$, $\mathbf{X}_{i+1} = \mathbf{X}_s$ and $\mathbf{X}_t$ is the non-identical point, then the plane $\mathbf{B}_i$ is linear in $\mathbf{X}_t$, since equation (2.154) can in this case also be written in the form

$$\begin{aligned} \mathbf{B}_i &= -\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_{i+1})^T \tilde{\mathsf{C}}^*_\infty \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_{i+1})^T \mathbf{X}_t & (2.157)\\ &= \mathsf{U}_i(\mathbf{X}_i, \mathbf{X}_{i+1})\mathbf{X}_t & (2.158) \end{aligned}$$

It is, however, not linear in $\mathbf{X}_i$ or $\mathbf{X}_{i+1}$. To obtain the required Jacobians tensor notation is useful and equation (2.157) is re-written as

$$\begin{aligned} B_g^{(i)} &= -\overline{\Gamma}_{fg}(\Pi_{hj}(X_k^{(i)})X_j^{(i+1)})\tilde{C}^*_{fb}\overline{\Gamma}_{ab}(\Pi_{cd}(X_e^{(i)})X_d^{(i+1)})X_a^{(t)} & (2.159)\\ &= -\overline{\Gamma}_{fg}(\Pi_{hj}(X_k^{(i+1)})X_j^{(i)})\tilde{C}^*_{fb}\overline{\Gamma}_{ab}(\Pi_{cd}(X_e^{(t)})X_d^{(i+1)})X_a^{(i)} & (2.160) \end{aligned}$$

where the indices of the points are put up eventually to improve the readability. Now the partial derivatives are easily obtained using product and chain rule as

$$\begin{aligned} \frac{\partial B_g^{(i)}}{\partial X_m^{(i)}} &= -\overline{\Gamma}'_{lfg}\Pi_{lm}(\mathbf{X}^{(i+1)})\tilde{C}^*_{fb}\overline{\Gamma}_{ab}(\mathbf{\Pi}(\mathbf{X}^{(t)})\mathbf{X}^{(i+1)})X_a^{(i)} & (2.161)\\ &\quad -\overline{\Gamma}_{fg}(\mathbf{\Pi}(\mathbf{X}^{(i+1)})\mathbf{X}^{(i)})\tilde{C}^*_{fb}\overline{\Gamma}_{mb}(\mathbf{\Pi}(\mathbf{X}^{(t)})\mathbf{X}^{(i+1)}) \end{aligned}$$

denoting the partial derivatives of the construction matrix with

$$\overline{\Gamma}'_{rst} = \frac{\partial \overline{\Gamma}_{st}}{\partial L_r} = \begin{cases} +1 & \text{if } \overline{\Gamma}_{st}(1,2,3,4,5,6) = r \\ -1 & \text{if } \overline{\Gamma}_{st}(1,2,3,4,5,6) = -r \\ 0 & \text{otherwise} \end{cases} \quad (2.162)$$

so that the Jacobian is

$$\mathsf{V}_i(\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{X}_t) = \left(\left. \frac{\partial B_g^{(i)}}{\partial X_m^{(i)}} \right|_{\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{X}_t} \right)_{g=1..4, m=1..4} \quad (2.163)$$

Because equation (2.157) is reflexive in $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$ also its derivatives are, so that the third Jacobian is simply obtained from the last by exchanging $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$, i.e.

$$\left(\left. \frac{\partial B_g^{(i)}}{\partial X_m^{(i+1)}} \right|_{\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{X}_t} \right)_{g=1..4, m=1..4} = \mathsf{V}_i(\mathbf{X}_{i+1}, \mathbf{X}_i, \mathbf{X}_t) \quad (2.164)$$

Hence, the covariance matrix of the delimiting plane is now computable as

$$\mathsf{C}_{B_iB_i} = \mathsf{J}_1 \begin{pmatrix} \mathsf{C}_{X_tX_t} & \mathsf{C}_{X_tX_i} & \mathsf{C}_{X_tX_{i+1}} \\ \mathsf{C}_{X_iX_t} & \mathsf{C}_{X_iX_i} & \mathsf{C}_{X_iX_{i+1}} \\ \mathsf{C}_{X_{i+1}X_t} & \mathsf{C}_{X_{i+1}X_i} & \mathsf{C}_{X_{i+1}X_{i+1}} \end{pmatrix} \mathsf{J}_1^T \quad (2.165)$$

using the Jacobian

$$\mathsf{J}_1(\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{X}_t) = \begin{pmatrix} \mathsf{U}_i(\mathbf{X}_i, \mathbf{X}_{i+1}) \\ \mathsf{V}_i(\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{X}_t) \\ \mathsf{V}_i(\mathbf{X}_{i+1}, \mathbf{X}_i, \mathbf{X}_t) \end{pmatrix}^T \quad (2.166)$$

Case 2: In the second case, assuming without loss of generality $\mathbf{X}_i = \mathbf{X}_r$ and neither $\mathbf{X}_s$ nor $\mathbf{X}_t$ identical to $\mathbf{X}_{i+1}$, equation (2.154) is linear in $\mathbf{X}_s$, $\mathbf{X}_t$ and $\mathbf{X}_{i+1}$, since it can be written either as

$$\begin{aligned}\mathbf{B}_i &= -\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_{i+1})^T\tilde{\mathsf{C}}^*_\infty\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_s)^T\mathbf{X}_t &(2.167)\\ &= \mathsf{U}_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_s)\mathbf{X}_t &(2.168)\\ &= -\mathsf{U}_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_t)\mathbf{X}_s &(2.169)\end{aligned}$$

or as

$$\begin{aligned}\mathbf{B}_i &= -\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\tilde{\mathsf{C}}^*_\infty\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_t)^T\mathbf{X}_s)^T\mathbf{X}_{i+1} &(2.170)\\ &= \mathsf{V}_i(\mathbf{X}_i,\mathbf{X}_s,\mathbf{X}_t)\mathbf{X}_{i+1} &(2.171)\end{aligned}$$

It is not linear in $\mathbf{X}_i$, though. Using again tensor notation, equation (2.167) is re-written as

$$\begin{aligned}B_g^{(i)} &= -\overline{\Gamma}_{fg}(\Pi_{hj}(X_k^{(i)})X_j^{(i+1)})\tilde{C}^*_{fb}\overline{\Gamma}_{ab}(\Pi_{cd}(X_e^{(i)})X_d^{(s)})X_a^{(t)} &(2.172)\\ &= -\overline{\Gamma}_{fg}(\Pi_{hj}(X_k^{(i+1)})X_j^{(i)})\tilde{C}^*_{fb}\overline{\Gamma}_{ab}(\Pi_{cd}(X_e^{(t)})X_d^{(s)})X_a^{(i)} &(2.173)\end{aligned}$$

Applying product and chain rule the partial derivatives are

$$\begin{aligned}\frac{\partial B_g^{(i)}}{\partial X_m^{(i)}} &= -\overline{\Gamma}'_{lfg}\Pi_{lm}(\mathbf{X}^{(i+1)})\tilde{C}^*_{fb}\overline{\Gamma}_{ab}(\mathbf{\Pi}(\mathbf{X}^{(t)})\mathbf{X}^{(s)})X_a^{(i)} &(2.174)\\ &\quad -\overline{\Gamma}_{fg}(\mathbf{\Pi}(\mathbf{X}^{(i+1)})\mathbf{X}^{(i)})\tilde{C}^*_{fb}\overline{\Gamma}_{mb}(\mathbf{\Pi}(\mathbf{X}^{(t)})\mathbf{X}^{(s)})\end{aligned}$$

so that the Jacobian is

$$\mathsf{W}_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_s,\mathbf{X}_t) = \left(\left.\frac{\partial B_g^{(i)}}{\partial X_m^{(i)}}\right|_{\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_s,\mathbf{X}_t}\right)_{g=1..4,m=1..4} \tag{2.175}$$

and the covariance matrix is obtained as

$$\mathsf{C}_{B_iB_i} = \mathsf{J}_2\begin{pmatrix}\mathsf{C}_{X_sX_s} & \mathsf{C}_{X_sX_t} & \mathsf{C}_{X_sX_i} & \mathsf{C}_{X_sX_{i+1}}\\ \mathsf{C}_{X_tX_s} & \mathsf{C}_{X_tX_t} & \mathsf{C}_{X_tX_i} & \mathsf{C}_{X_tX_{i+1}}\\ \mathsf{C}_{X_iX_s} & \mathsf{C}_{X_iX_t} & \mathsf{C}_{X_iX_i} & \mathsf{C}_{X_iX_{i+1}}\\ \mathsf{C}_{X_{i+1}X_s} & \mathsf{C}_{X_{i+1}X_t} & \mathsf{C}_{X_{i+1}X_i} & \mathsf{C}_{X_{i+1}X_{i+1}}\end{pmatrix}\mathsf{J}_2^T \tag{2.176}$$

using the Jacobian

$$\mathsf{J}_2(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_s,\mathbf{X}_t) = \begin{pmatrix}-\mathsf{U}_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_t)\\ \mathsf{U}_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_s)\\ \mathsf{W}_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_s,\mathbf{X}_t)\\ \mathsf{V}_i(\mathbf{X}_i,\mathbf{X}_s,\mathbf{X}_t)\end{pmatrix}^T \tag{2.177}$$

Case 3: Finally in the third case equation (2.154) is linear in all quantities $\mathbf{X}_i$, $\mathbf{X}_{i+1}$, $\mathbf{X}_r$, $\mathbf{X}_s$ and $\mathbf{X}_t$, as it can be written either as

$$\begin{aligned}\mathbf{B}_i &= -\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_{i+1})^T\tilde{\mathsf{C}}^*_\infty\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_r)\mathbf{X}_s)^T\mathbf{X}_t &(2.178)\\ &= \mathsf{U}_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_r,\mathbf{X}_s)\mathbf{X}_t &(2.179)\\ &= \mathsf{U}_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_t,\mathbf{X}_r)\mathbf{X}_s &(2.180)\\ &= \mathsf{U}_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_s,\mathbf{X}_t)\mathbf{X}_r &(2.181)\end{aligned}$$

or as

$$\begin{aligned}\mathbf{B}_i &= -\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\tilde{C}^*_\infty\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_r)\mathbf{X}_t)^T\mathbf{X}_s)^T\mathbf{X}_{i+1} && (2.182)\\ &= V_i(\mathbf{X}_i,\mathbf{X}_r,\mathbf{X}_s,\mathbf{X}_t)\mathbf{X}_{i+1} && (2.183)\\ &= -V_i(\mathbf{X}_{i+1},\mathbf{X}_r,\mathbf{X}_s,\mathbf{X}_t)\mathbf{X}_i && (2.184)\end{aligned}$$

Hence, the covariance matrix of the delimiting plane is in this case given by

$$\mathsf{C}_{B_iB_i} = J_3\begin{pmatrix}\mathsf{C}_{X_rX_r} & \mathsf{C}_{X_rX_s} & \mathsf{C}_{X_rX_t} & \mathsf{C}_{X_rX_i} & \mathsf{C}_{X_rX_{i+1}}\\ \mathsf{C}_{X_sX_r} & \mathsf{C}_{X_sX_s} & \mathsf{C}_{X_sX_t} & \mathsf{C}_{X_sX_i} & \mathsf{C}_{X_sX_{i+1}}\\ \mathsf{C}_{X_tX_r} & \mathsf{C}_{X_tX_s} & \mathsf{C}_{X_tX_t} & \mathsf{C}_{X_tX_i} & \mathsf{C}_{X_tX_{i+1}}\\ \mathsf{C}_{X_iX_r} & \mathsf{C}_{X_iX_s} & \mathsf{C}_{X_iX_t} & \mathsf{C}_{X_iX_i} & \mathsf{C}_{X_iX_{i+1}}\\ \mathsf{C}_{X_{i+1}X_r} & \mathsf{C}_{X_{i+1}X_s} & \mathsf{C}_{X_{i+1}X_t} & \mathsf{C}_{X_{i+1}X_i} & \mathsf{C}_{X_{i+1}X_{i+1}}\end{pmatrix}J_3^T \qquad (2.185)$$

using the Jacobian

$$J_3(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_r,\mathbf{X}_s,\mathbf{X}_t) = \begin{pmatrix}U_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_s,\mathbf{X}_t)\\ U_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_t,\mathbf{X}_r)\\ U_i(\mathbf{X}_i,\mathbf{X}_{i+1},\mathbf{X}_r,\mathbf{X}_s)\\ -V_i(\mathbf{X}_{i+1},\mathbf{X}_r,\mathbf{X}_s,\mathbf{X}_t)\\ V_i(\mathbf{X}_i,\mathbf{X}_r,\mathbf{X}_s,\mathbf{X}_t)\end{pmatrix}^T \qquad (2.186)$$

The computation of the covariance matrices between the entities is possible in a straightforward manner using the Jacobians derived. However, one must distinguish between all combinations of cases that can occur, so that this derivation is not presented here, as the covariance matrices between the entities are not required for the hypothesis tests.

Converting from plane to point representation: To convert a space polygon from plane representation into line representation, one has to intersect the three planes going through each corner point. Hence the corner points are (cf. equation (2.104))

$$\begin{aligned}\mathbf{X}_i &= \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{A})\mathbf{B}_{i-1})^T\mathbf{B}_i && (2.187)\\ &= \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B}_i)\mathbf{A})^T\mathbf{B}_{i-1} && (2.188)\\ &= \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B}_{i-1})\mathbf{B}_i)^T\mathbf{A} && (2.189)\end{aligned}$$

where the indices again read modulo N.

The covariance matrices are therefore obtained using the Jacobian

$$J = \begin{pmatrix}\overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B}_N)\mathbf{B}_1)^T & \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{A})\mathbf{B}_N)^T & & \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B}_1)\mathbf{A})^T\\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B}_1)\mathbf{B}_2)^T & \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B}_2)\mathbf{A})^T & \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{A})\mathbf{B}_1)^T & \\ \vdots & & \ddots & \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B}_{N-1})\mathbf{B}_N)^T & & \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{B}_N)\mathbf{A})^T & \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{A})\mathbf{B}_{N-1})^T\end{pmatrix} \qquad (2.190)$$

as

$$\left(\mathsf{C}_{X_iX_j}\right)_{i,j=1,\ldots,N} = J\begin{pmatrix}\mathsf{C}_{AA} & \mathsf{C}_{AB_1} & \cdots & \mathsf{C}_{AB_N}\\ \mathsf{C}_{B_1A} & \mathsf{C}_{B_1B_1} & \cdots & \mathsf{C}_{B_1B_N}\\ \vdots & \vdots & \ddots & \vdots\\ \mathsf{C}_{B_NA} & \mathsf{C}_{B_NB_1} & \cdots & \mathsf{C}_{B_NB_N}\end{pmatrix}J^T \qquad (2.191)$$

Converting from point to line representation: To convert a space polygon from point to line representation, one has to connect the adjacent corner points. The lines are then given by (cf. equation (2.83))

$$\mathbf{L}_i = \mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_{i+1} = -\mathbf{\Pi}(\mathbf{X}_{i+1})\mathbf{X}_i \qquad (2.192)$$

where the indices read modulo N. The covariance matrices are obtained using the Jacobian

$$\mathsf{J}(\mathbf{X}_1,...,\mathbf{X}_N) = \begin{pmatrix} -\mathbf{\Pi}(\mathbf{X}_2) & \mathbf{\Pi}(\mathbf{X}_1) & & \\ & \ddots & & \\ & & -\mathbf{\Pi}(\mathbf{X}_N) & \mathbf{\Pi}(\mathbf{X}_{N-1}) \\ \mathbf{\Pi}(\mathbf{X}_N) & & & -\mathbf{\Pi}(\mathbf{X}_1) \end{pmatrix} \tag{2.193}$$

as

$$\left(\mathsf{C}_{L_iL_j}\right)_{i,j=1,...,N} = J(\mathbf{X}_1,...,\mathbf{X}_N)\left(\mathsf{C}_{X_iX_j}\right)_{i,j=1,...,N} J(\mathbf{X}_1,...,\mathbf{X}_N)^T \tag{2.194}$$

This is completely analogous to the 2d case.

Converting from line to point representation: To convert a 3d polygon from line to point representation, one has to intersect all adjacent lines. As before in the conversion from point to plane representation, this is not possible in a unique manner, since the uncertainties of the adjacent lines is not guaranteed to fit at the intersections. The point $\mathbf{X}_i$ is at the intersection of $\mathbf{L}_{i-1}$ and $\mathbf{L}_i$, which exists because of condition (2.151). Hence, neither the plane constructed from $\mathbf{L}_{i-1}$ and $\mathbf{X}_i$ nor from from $\mathbf{L}_i$ and $\mathbf{X}_i$ does exists (cf. equation 2.95), i. e.

$$\overline{\mathbf{\Gamma}}(\mathbf{L}_{i-1})^T\mathbf{X}_i = \begin{pmatrix} \overline{\boldsymbol{\gamma}}_1^{(i-1)} \\ \vdots \\ \overline{\boldsymbol{\gamma}}_4^{(i-1)} \end{pmatrix}\mathbf{X}_i = \mathbf{0} \tag{2.195}$$

and

$$\overline{\mathbf{\Gamma}}(\mathbf{L}_{i})^T\mathbf{X}_i = \begin{pmatrix} \overline{\boldsymbol{\gamma}}_1^{(i)} \\ \vdots \\ \overline{\boldsymbol{\gamma}}_4^{(i)} \end{pmatrix}\mathbf{X}_i = \mathbf{0} \tag{2.196}$$

This may also be interpreted as $\mathbf{X}_i$ lying on the eight planes $\overline{\boldsymbol{\gamma}}_1^{(i-1)}$, ..., $\overline{\boldsymbol{\gamma}}_4^{(i-1)}$, $\overline{\boldsymbol{\gamma}}_1^{(i)}$, ..., $\overline{\boldsymbol{\gamma}}_4^{(i)}$ defined by the columns of the two $\overline{\mathbf{\Gamma}}$-matrices.

As a point is uniquely defined by three planes and the rank of each $\overline{\mathbf{\Gamma}}$-matrix is two, only three of those rows and two of each $\overline{\mathbf{\Gamma}}$-matrix are linear independent. Thus selecting without loss of generality three of those columns $\overline{\boldsymbol{\gamma}}_r^{(i-1)}$, $\overline{\boldsymbol{\gamma}}_s^{(i-1)}$ and $\overline{\boldsymbol{\gamma}}_t^{(i)}$, either two of the first and one of the second $\overline{\mathbf{\Gamma}}$-matrix or vice versa, the point is obtained by intersecting the three planes (cf. equation (2.102))

$$\mathbf{X}_i = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\overline{\boldsymbol{\gamma}}_r^{(i-1)})\overline{\boldsymbol{\gamma}}_s^{(i-1)})^T\overline{\boldsymbol{\gamma}}_t^{(i)} \tag{2.197}$$

This expression is linear in $\mathbf{L}^{(i)}$, but not in $\boldsymbol{L}^{(i-1)}$. To obtain the covariance matrix, it is re-written in tensor notation as

$$\begin{aligned} X_b^{(i)} &= \overline{\Gamma}_{ab}(\Pi_{cd}(\overline{\Gamma}_{re}(\mathbf{L}^{(i-1)}))\overline{\Gamma}_{sd}(\mathbf{L}^{(i-1)}))\overline{\Gamma}_{ta}(\mathbf{L}^{(i)}) & (2.198)\\ &= \overline{\Gamma}_{ab}(\Pi_{cd}(\overline{\Gamma}_{te}(\mathbf{L}^{(i)}))\overline{\Gamma}_{rd}(\mathbf{L}^{(i-1)}))\overline{\Gamma}_{sa}(\mathbf{L}^{(i-1)}) & (2.199)\end{aligned}$$

Applying product and chain rule, the partial derivatives are

$$\frac{\partial X_b^{(i)}}{\partial L_f^{(i)}} = \overline{\Gamma}_{ab}(\Pi_{cd}(\overline{\Gamma}_{re}(\mathbf{L}^{(i-1)}))\overline{\Gamma}_{sd}(\mathbf{L}^{(i-1)}))\overline{\Gamma}'_{fta} \tag{2.200}$$

and

$$\begin{aligned} \frac{\partial X_b^{(i)}}{\partial L_f^{(i+1)}} &= \overline{\Gamma}_{ab}(\Pi_{cd}(\overline{\Gamma}_{te}(\mathbf{L}^{(i)}))\overline{\Gamma}_{rd}(\mathbf{L}^{(i-1)}))\overline{\Gamma}'_{fsa} & (2.201)\\ &\quad +\overline{\Gamma}'_{gab}\Pi_{gh}(\overline{\Gamma}_{tk}(\mathbf{L}^{(i)}))\overline{\Gamma}'_{frh}\overline{\Gamma}_{sa}(\mathbf{L}^{(i-1)}) \end{aligned}$$

so that the Jacobians are given by

$$\mathsf{V}_i(\mathbf{L}_{i-1}) = \left(\left. \frac{\partial X_b^{(i)}}{\partial L_f^{(i)}} \right|_{\mathbf{L}_{i-1}} \right)_{b=1..4,f=1..4} \tag{2.202}$$

and

$$\mathsf{W}_i(\mathbf{L}_{i-1}, \mathbf{L}_i) = \left(\left. \frac{\partial X_b^{(i)}}{\partial L_f^{(i-1)}} \right|_{\mathbf{L}_{i-1},\mathbf{L}_i} \right)_{b=1..4,f=1..4} \tag{2.203}$$

Finally the covariance matrices are obtained using the Jacobian

$$\mathsf{J}(\mathbf{L}_1, ..., \mathbf{L}_N) = \begin{pmatrix} \mathsf{V}_1(\mathbf{L}_N) & & & \mathsf{W}_1(\mathbf{L}_N, \mathbf{L}_1) \\ \mathsf{W}_2(\mathbf{L}_1, \mathbf{L}_2) & \mathsf{V}_2(\mathbf{L}_1) & & \\ & \ddots & & \\ & & \mathsf{W}_N(\mathbf{L}_{N-1}, \mathbf{L}_N) & \mathsf{V}_N(\mathbf{L}_{N-1}) \end{pmatrix} \tag{2.204}$$

as

$$\left(\mathsf{C}_{X_i X_j}\right)_{i,j=1,...,N} = \mathsf{J}(\mathbf{L}_1, ..., \mathbf{L}_N) \left(\mathsf{C}_{L_i L_j}\right)_{i,j=1,...,N} \mathsf{J}(\mathbf{L}_1, ..., \mathbf{L}_N)^T \tag{2.205}$$

Up to now a number of 2d and 3d entities have been presented. It has been shown, how to represent and convert the various entities in 2d into each other as well as how to represent and convert the various entities in 3d into each other. The connection between 2d and 3d space, i.e. how to obtain 2d entities from 3d entities and vice versa, has not been discussed, so far. In computer vision and photogrammetry the connection between 2d and 3d space is established by projective cameras, which are also representable in the uncertain oriented projective framework and will therefore be presented in the following.

2.1.5 Uncertain oriented projective cameras

In this work, cameras are seen as devices that relate the uncertain oriented projective 2d space with the uncertain oriented projective 3d space. Furthermore, projective cameras can be represented as uncertain oriented projective 3d entities themselves and therefore fit into the framework as such entities. Like many of the entities discussed above, also projective cameras can be represented in two representations, namely in point projection matrix representation and in line projection matrix representation. In the following, those two representations in the uncertain oriented projective framework will be presented and it will be shown, how those uncertain oriented projective cameras relate the 3d object space to the 2d image space and vice versa. Especially the possibility of relating the 2d image entities to 3d scene entities will enable the reconstruction application presented in chapter 4.

Point projection matrix representation

The first and most well-known representation for a projective camera is given by the point projection matrix. In the following the point projection matrix will be presented and it will be shown, how it can be represented as uncertain oriented projective 3d entity.

In section 1.3.1 various methods for calibrating cameras are presented. Omitting non-linear image distortions those algorithms usually output five calibration parameters: the principal length c, the principal point (x_h, y_h), the aspect ration m and the coordinate axis

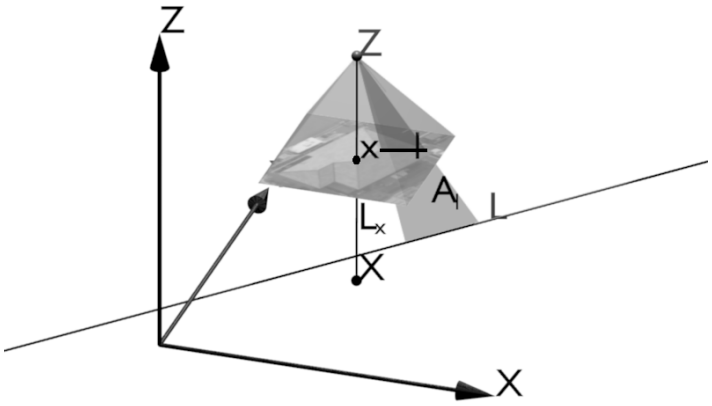

Figure 2.11: A line preserving projective camera images the scene line $\mathbf{L}$ as image line $\mathbf{l}$ and the scene point $\mathbf{X}$ as image point $\mathbf{x}$. The backward projection is possible, too. The image point $\mathbf{x}$ backprojects onto the scene line $\mathbf{L}_x$ and the image line $\mathbf{l}$ backprojects onto the scene plane $\mathbf{A}_l$.

skew s. Those parameters are then collected in the homogeneous calibration matrix (cf. [Mugnier *et al.*, 2004, p.224])

$$\mathsf{K} = \begin{pmatrix} c & cs & x_h \\ 0 & c(1+m) & y_h \\ 0 & 0 & 1 \end{pmatrix} \tag{2.206}$$

that describes, how directions of incoming light are mapped to pixel positions in the image. In the oriented projective setting, light rays coming from the front of the camera result in 2d image points with positive orientation and light rays coming from the rear of the camera result in 2d image points with negative orientation. It is therefore clear that all visible points in an image take with a real physical camera must have positive orientation.

Furthermore in section 1.3.2 methods for estimating the position $\boldsymbol{Z}$ and rotation matrix R of the camera in space are presented. The homogeneous point projection matrix is then given by (cf. [Mugnier *et al.*, 2004, p.225] and [Hartley and Zisserman, 2000, p.143])

$$\mathsf{P} = \mathsf{K}\mathsf{R}(I_3| - \boldsymbol{Z}) \tag{2.207}$$

As this matrix is uniquely defined up to a positive scale factor, its elements can be collected into a vector in oriented projective space

$$\mathbf{p} = \mathrm{vec}\mathsf{P}^T \tag{2.208}$$

so that the resulting uncertain oriented projective entity

$$(\mathbf{p}, \mathsf{C}_{pp}) \tag{2.209}$$

represents a line preserving projective camera. The elements of the covariance matrix C_{pp} can be obtained from the uncertainties of the calibration and orientation parameters by error

propagation. As this depends on the specific representation of those, this will not be discussed here and the covariance matrix is assumed to be given by the orientation procedure.

Given an uncertain oriented projective 3d point $(\mathbf{X}, \mathsf{C}_{XX})$ the projection into the uncertain oriented projective 2d image point is given by (cf. [Hartley and Zisserman, 2000, p.142] and see figure 2.11)

$$\mathbf{x} = \mathsf{P}\mathbf{X} \tag{2.210}$$

having the covariance matrix (cf. [Koch, 1997, p.44])

$$\mathsf{C}_{xx} = \mathsf{P}\mathsf{C}_{XX}\mathsf{P}^T + (I_3 \otimes \mathbf{X}^T)\mathsf{C}_{pp}(I_3 \otimes \mathbf{X}^T)^T \tag{2.211}$$

Observe that the orientation is reversed for points behind the camera.

On the other hand, given an uncertain oriented projective image line $(\mathbf{l}, \mathsf{C}_{ll})$, the projecting uncertain oriented scene plane is given by (cf. [Hartley and Zisserman, 2000, p.186] and see figure 2.11)

$$\mathbf{A}_l = \mathsf{P}^T\mathbf{l} \tag{2.212}$$

having the covariance matrix

$$\mathsf{C}_{A_l A_l} = \mathsf{P}^T C_{ll}\mathsf{P} + (\mathbf{l}^T \otimes I_4)\mathsf{C}_{pp}(\mathbf{l}^T \otimes I_4)^T \tag{2.213}$$

It has been seen that the point projection matrix representation of a projective camera is well-suited for computing the uncertain oriented projective 2d image point from an uncertain oriented projective 3d scene point and for computing the uncertain oriented projective 3d scene plane from an uncertain oriented projective 2d image line. In the following another representation will be presented that is better suited for transferring 3d lines into images and more important 2d image points into 3d scene lines.

Line projection matrix representation

The second useful representation for a projective camera is the line projection matrix (cf. [Mugnier *et al.*, 2004, p.236])

$$\mathsf{Q} = (\mathsf{K}R)^{-1}\left(-S(\boldsymbol{Z})|I_3\right) \tag{2.214}$$

which defines the oriented projective entity

$$\mathbf{q} = \mathrm{vec}\mathsf{Q}^T \tag{2.215}$$

and, adding a covariance matrix, the uncertain oriented projective entity

$$(\mathbf{q}, \mathsf{C}_{qq}) \tag{2.216}$$

Obviously, as the projective camera has only eleven degrees of freedom, not all such 18-vectors actually represent cameras. Those constraints can be derived by noting that the rows of Q represent three lines through the projection center $\boldsymbol{Z}$ (cf. [Faugeras and Luong, 2001, p.195]). Therefore it is required that the rows fulfill the Plücker constraint and that all rows intersect in a single point, yielding a total of six constraints. Hence, a matrix

$$\mathsf{Q} = \begin{pmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \mathbf{q}_3^T \end{pmatrix} \tag{2.217}$$

represents a projective camera if and only if the six conditions

$$\mathbf{q}_1^T D_6 \mathbf{q}_1 = 0 \tag{2.218}$$

$$\mathbf{q}_2^T D_6 \mathbf{q}_2 = 0 \tag{2.219}$$

$$\mathbf{q}_3^T D_6 \mathbf{q}_3 = 0 \tag{2.220}$$

$$\mathbf{q}_1^T D_6 \mathbf{q}_2 = 0 \tag{2.221}$$

$$\mathbf{q}_1^T D_6 \mathbf{q}_3 = 0 \tag{2.222}$$

$$\mathbf{q}_2^T D_6 \mathbf{q}_3 = 0 \tag{2.223}$$

are fulfilled.

Given an uncertain oriented projective 3d line $(\mathbf{L}, \mathsf{C}_{LL})$ the projection into the uncertain oriented projected 2d image line is given by (see figure 2.11)

$$\mathbf{l} = \mathsf{Q}\mathbf{L} \tag{2.224}$$

having the covariance matrix

$$\mathsf{C}_{ll} = \mathsf{Q}\mathsf{C}_{LL}\mathsf{Q}^T + (I_6 \otimes \mathbf{L}^T)\mathsf{C}_{qq}(I_6 \otimes \mathbf{L}^T)^T \tag{2.225}$$

On the other hand given an uncertain oriented image point $(\mathbf{x}, \mathsf{C}_{xx})$, the projecting uncertain oriented scene line is given by (see figure 2.11)

$$\mathbf{L}_x = D_6 \mathsf{Q}^T \mathbf{x} \tag{2.226}$$

having the covariance matrix

$$\mathsf{C}_{L_x L_x} = D_6 \mathsf{Q}^T C_{xx} \mathsf{Q} D_6 + (\mathbf{x}^T \otimes D_6)\mathsf{C}_{qq}(\mathbf{x}^T \otimes D_6)^T \tag{2.227}$$

The line projection matrix representation, again representable as uncertain oriented projective 3d entity, turned out to be very well-suited for the task of transferring uncertain oriented projective 3d scene lines into uncertain oriented projective 2d image lines and also uncertain oriented projective 2d image points into uncertain oriented projective 3d lines. As all those operations are often required, the two representations need to be convertible. In the following it will be shown, how an uncertain oriented projective camera in point projection matrix representation can be converted into an uncertain oriented projective camera in line projection matrix representation.

Converting from point projection to line projection representation

It is possible to convert a point projection matrix into a line projection matrix without having to decompose it, which significantly eases uncertainty propagation, as will be seen below. However, the conversion in the other direction is not easily possible without decomposition. Fortunately it is not required in many applications and will not be discussed here.

Now given a point projection matrix

$$\mathsf{P} = \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{pmatrix} \tag{2.228}$$

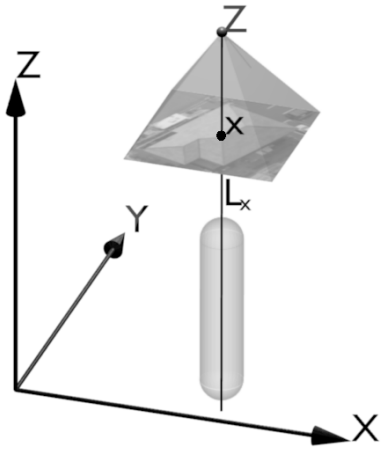

Figure 2.12: If a minimal and maximal distance from the camera are known, an image point $\mathbf{x}$ backprojects onto an uncertain oriented line segment in space.

its rows $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ are the planes of the camera coordinate system. To obtain the rows of the line projection matrix one has to intersect those planes yielding (cf. equation (2.109))

$$\mathsf{Q} = \begin{pmatrix} \left(\overline{\mathbf{\Pi}}(\mathbf{p}_2)\mathbf{p}_3\right)^T \\ \left(\overline{\mathbf{\Pi}}(\mathbf{p}_3)\mathbf{p}_1\right)^T \\ \left(\overline{\mathbf{\Pi}}(\mathbf{p}_1)\mathbf{p}_2\right)^T \end{pmatrix} \tag{2.229}$$

The Jacobian of this non-linear transformation is (cf. [Heuel, 2004, p.59])

$$J_{qp}(\mathbf{p}) = \frac{\partial \mathbf{q}}{\partial \mathbf{p}} = \begin{pmatrix} 0 & -\overline{\mathbf{\Pi}}(\mathbf{p}_3) & \overline{\mathbf{\Pi}}(\mathbf{p}_2) \\ \overline{\mathbf{\Pi}}(\mathbf{p}_3) & 0 & -\overline{\mathbf{\Pi}}(\mathbf{p}_1) \\ -\overline{\mathbf{\Pi}}(\mathbf{p}_2) & \overline{\mathbf{\Pi}}(\mathbf{p}_1) & 0 \end{pmatrix} \tag{2.230}$$

so that the covariance matrix transforms according to

$$\mathsf{C}_{qq} = J_{qp}(\mathbf{p})\mathsf{C}_{pp}J_{qp}(\mathbf{p})^T \tag{2.231}$$

Up to now two representations for projective cameras, the conversion between those representations and the forward and backward projection of points and lines have been presented. In many applications lower and upper bounds on the distance of the object from the cameras are available that enable the backprojection of 2d image points into 3d line segments instead of lines and the backprojection of 2d image line segments into 3d polygons instead of planes. In the following this two specific backprojections will be discussed.

Backward projection of points

It has been shown in equation (2.226), how the projecting line of an image point can be computed. This line extends from the camera center in both directions toward infinity. Obviously the true scene point cannot be behind the camera, which could be modeled in an oriented projective framework (cf. [Hartley, 1998] and [Hartley and Zisserman, 2000,

p.149]). Furthermore many applications allow to specify a minimal possible distance $d_{\min}$ and a maximal possible distance $d_{\max}$ of the object from the camera. Therefore the locus of an image point in space will be modeled as an uncertain oriented projective 3d line segment in the following. The construction of the line segment is as follows: Given the uncertain oriented projective image point

$$(\mathbf{x}, \mathsf{C}_{xx}) \tag{2.232}$$

and the uncertain oriented projective camera in point projection representation

$$(\mathbf{p}, \mathsf{C}_{pp}) \tag{2.233}$$

as well as in line projection representation

$$(\mathbf{q}, \mathsf{C}_{qq}) \tag{2.234}$$

one first constructs the line joining the projection center and the image point according to equation (2.226)

$$\begin{pmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_0 \end{pmatrix} = \mathsf{D}_6 \mathsf{Q}^T \mathbf{x} \tag{2.235}$$

It has the covariance matrix

$$\begin{pmatrix} \mathsf{C}_{L_h L_h} & \mathsf{C}_{L_h L_0} \\ \mathsf{C}_{L_h L_0} & \mathsf{C}_{L_0 L_0} \end{pmatrix} = \mathsf{D}_6 \mathsf{Q}^T C_{xx} \mathsf{Q} \mathsf{D}_6 + (\mathbf{x}^T \otimes \mathsf{D}_6) \mathsf{C}_{qq} (\mathbf{x}^T \otimes \mathsf{D}_6)^T) \tag{2.236}$$

Now the projection center is derived from the point projection matrix as being the intersection of all three coordinate planes represented by its rows. It is given by (cf. equation (2.104))

$$\mathbf{Z} = \begin{pmatrix} \boldsymbol{Z}_0 \\ Z_h \end{pmatrix} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_1)\mathbf{p}_2)^T \mathbf{p}_3 \tag{2.237}$$

having the covariance matrix

$$\mathsf{C}_{ZZ} = \begin{pmatrix} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_2)\mathbf{p}_3) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_3)\mathbf{p}_1) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_1)\mathbf{p}_2) \end{pmatrix}^T \mathsf{C}_{pp} \begin{pmatrix} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_2)\mathbf{p}_3) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_3)\mathbf{p}_1) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_1)\mathbf{p}_2) \end{pmatrix} \tag{2.238}$$

The two Euclidean points

$$\boldsymbol{X} = \frac{\boldsymbol{Z}_0}{Z_h} + d_{\min} \frac{\boldsymbol{L}_h}{|\boldsymbol{L}_h|} \tag{2.239}$$

and

$$\boldsymbol{Y} = \frac{\boldsymbol{Z}_0}{Z_h} + d_{\max} \frac{\boldsymbol{L}_h}{|\boldsymbol{L}_h|} \tag{2.240}$$

now constitute the two end-points of the line segment. Under the assumption that the direction of the projecting ray is uncorrelated with the position of the camera, the Euclidean covariance matrices of the two end-points are obtained using the Jacobian (cf. equation (2.79) and equation (2.14))

$$J = \begin{pmatrix} \frac{1}{Z_h} I_3 & \frac{-\boldsymbol{Z}_0}{Z_h^2} & \frac{d_{\min}}{|\boldsymbol{L}_h|} \left(I_3 - \frac{\boldsymbol{L}_h \boldsymbol{L}_h^T}{|\boldsymbol{L}_h|^2} \right) \\ \frac{1}{Z_h} I_3 & \frac{-\boldsymbol{Z}_0}{Z_h^2} & \frac{d_{\max}}{|\boldsymbol{L}_h|} \left(I_3 - \frac{\boldsymbol{L}_h \boldsymbol{L}_h^T}{|\boldsymbol{L}_h|^2} \right) \end{pmatrix} \tag{2.241}$$

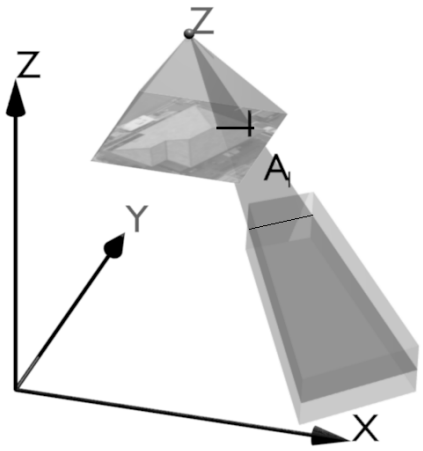

Figure 2.13: If a minimal and maximal distance from the camera are known, an image line $\mathbf{l}$ backprojects onto an uncertain oriented polygon in space.

as

$$\begin{pmatrix} \mathsf{C}_{XX} & \mathsf{C}_{XY} \\ \mathsf{C}_{YX} & \mathsf{C}_{YY} \end{pmatrix} = J \begin{pmatrix} \mathsf{C}_{ZZ} & 0 \\ 0 & \mathsf{C}_{L_h L_h} \end{pmatrix} J^T \tag{2.242}$$

The uncertain oriented projective end-points are easily obtained from the uncertain Euclidean points as described in section 2.1.3 and the resulting uncertain oriented projective 3d line segment (see figure 2.12) in point representation is given by

$$(\{\mathbf{X}, \mathbf{Y}\}, \{\mathsf{C}_{XX}, \mathsf{C}_{YY}, \mathsf{C}_{XY}\}) \tag{2.243}$$

If two such line segments from two distinct cameras intersect in space, then the image points can be matched with each other. This is a stronger condition than the pure epipolar constraint, as the distances of the scene point from the cameras are taken into account, too. In the section 2.2 a method will be presented, how this intersection between two uncertain oriented projective line segments can be tested.

Backward projection of line segments

How to compute the infinite projecting plane of an infinite image line has been shown in equation (2.212). This section deals with the backward projection of image line segments. As in the previous section, a minimal possible distance $d_{\min}$ and a maximal possible distance $d_{\max}$ of the object from the camera is assumed, so that the resulting locus of the image line segment is a space polygon with four corners. Given the uncertain oriented projective image line segment in point representation

$$(\{\mathbf{x}, \mathbf{y}\}, \{\mathsf{C}_{xx}, \mathsf{C}_{yy}, \mathsf{C}_{xy}\}) \tag{2.244}$$

as well as in line representation

$$(\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}, \{\mathsf{C}_{ll}, \mathsf{C}_{mm}, \mathsf{C}_{nn}, \mathsf{C}_{lm}, \mathsf{C}_{ln}, \mathsf{C}_{mn}\}) \tag{2.245}$$

together with the uncertain oriented projective camera in point projection representation

$$(\mathbf{p}, \mathsf{C}_{pp}) \tag{2.246}$$

as well as in line projection representation

$$(\mathbf{q}, \mathsf{C}_{qq}) \tag{2.247}$$

one is able to directly construct the two delimiting lines (cf. equation (2.226))

$$\mathbf{L}_1 = D_6 \mathsf{Q}^T \mathbf{x} \tag{2.248}$$

and

$$\mathbf{L}_3 = D_6 \mathsf{Q}^T \mathbf{y} \tag{2.249}$$

Their covariance matrices are obtainable using the Jacobian

$$J_a = \begin{pmatrix} D_6 \mathsf{Q}^T & 0 & \mathbf{x}^T \otimes D_6 \\ 0 & D_6 \mathsf{Q}^T & \mathbf{y}^T \otimes D_6 \end{pmatrix} \tag{2.250}$$

as

$$\begin{pmatrix} \mathsf{C}_{L_1 L_1} & \mathsf{C}_{L_1 L_3} \\ \mathsf{C}_{L_3 L_1} & \mathsf{C}_{L_3 L_3} \end{pmatrix} = J_a \begin{pmatrix} \mathsf{C}_{xx} & \mathsf{C}_{xy} & 0 \\ \mathsf{C}_{yx} & \mathsf{C}_{yy} & 0 \\ 0 & 0 & \mathsf{C}_{qq} \end{pmatrix} J_a^T \tag{2.251}$$

Furthermore the projecting plane is given by (cf. equation (2.212))

$$\mathbf{A} = \mathsf{P}^T \mathbf{l} \tag{2.252}$$

as well as the two delimiting planes

$$\mathbf{B}_1 = \mathsf{P}^T \mathbf{m} \tag{2.253}$$

and

$$\mathbf{B}_3 = \mathsf{P}^T \mathbf{n} \tag{2.254}$$

Their covariance matrices are obtained using the Jacobian

$$J_b = \begin{pmatrix} \mathsf{P}^T & 0 & 0 & \mathbf{l}^T \otimes I_4 \\ 0 & \mathsf{P}^T & 0 & \mathbf{m}^T \otimes I_4 \\ 0 & 0 & \mathsf{P}^T & \mathbf{n}^T \otimes I_4 \end{pmatrix} \tag{2.255}$$

as

$$\begin{pmatrix} \mathsf{C}_{AA} & \mathsf{C}_{AB_1} & \mathsf{C}_{AB_2} \\ \mathsf{C}_{B_1 A} & \mathsf{C}_{B_1 B_1} & \mathsf{C}_{B_1 B_2} \\ \mathsf{C}_{B_2 A} & \mathsf{C}_{B_2 B_1} & \mathsf{C}_{B_2 B_2} \end{pmatrix} = J_b \begin{pmatrix} \mathsf{C}_{ll} & \mathsf{C}_{lm} & \mathsf{C}_{ln} & 0 \\ \mathsf{C}_{ml} & \mathsf{C}_{mm} & \mathsf{C}_{mn} & 0 \\ \mathsf{C}_{nl} & \mathsf{C}_{nm} & \mathsf{C}_{nn} & 0 \\ 0 & 0 & 0 & \mathsf{C}_{pp} \end{pmatrix} J_b^T \tag{2.256}$$

As already shown in the previous section the projection center of the camera is given by

$$\mathbf{Z} = \begin{pmatrix} \boldsymbol{Z}_0 \\ Z_h \end{pmatrix} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_1)\mathbf{p}_2)^T \mathbf{p}_3 \tag{2.257}$$

having the covariance matrix

$$\mathsf{C}_{ZZ} = \begin{pmatrix} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_2)\mathbf{p}_3) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_3)\mathbf{p}_1) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_1)\mathbf{p}_2) \end{pmatrix}^T \mathsf{C}_{pp} \begin{pmatrix} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_2)\mathbf{p}_3) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_3)\mathbf{p}_1) \\ \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{p}_1)\mathbf{p}_2) \end{pmatrix} \tag{2.258}$$

As before the four corner-points of the uncertain oriented projective 3d polygon representing the image line segment are obtained from this, the lines $\mathbf{L}_1$ and $\mathbf{L}_3$ as well as the minimal possible distance $d_{\min}$ and the maximal possible distance $d_{\max}$. This yields the four Euclidean points

$$\boldsymbol{X}_1 = \frac{\boldsymbol{Z}_0}{Z_h} + d_{\min}\frac{\boldsymbol{L}_h^{(1)}}{|\boldsymbol{L}_h^{(1)}|} \tag{2.259}$$

$$\boldsymbol{X}_2 = \frac{\boldsymbol{Z}_0}{Z_h} + d_{\min}\frac{\boldsymbol{L}_h^{(3)}}{|\boldsymbol{L}_h^{(3)}|} \tag{2.260}$$

$$\boldsymbol{X}_3 = \frac{\boldsymbol{Z}_0}{Z_h} + d_{\max}\frac{\boldsymbol{L}_h^{(3)}}{|\boldsymbol{L}_h^{(3)}|} \tag{2.261}$$

and

$$\boldsymbol{X}_4 = \frac{\boldsymbol{Z}_0}{Z_h} + d_{\max}\frac{\boldsymbol{L}_h^{(1)}}{|\boldsymbol{L}_h^{(1)}|} \tag{2.262}$$

where the indices have been put up eventually to improve readability. Using the Jacobian

$$J = \begin{pmatrix} \frac{1}{Z_h}I_3 & \frac{-\boldsymbol{Z}_0}{Z_h^2} & \frac{d_{\min}}{\left|\boldsymbol{L}_h^{(1)}\right|}\left(I_3 - \frac{\boldsymbol{L}_h^{(1)}\boldsymbol{L}_h^{(1)^T}}{\left|\boldsymbol{L}_h^{(1)}\right|^2}\right) & 0 \\ \frac{1}{Z_h}I_3 & \frac{-\boldsymbol{Z}_0}{Z_h^2} & 0 & \frac{d_{\min}}{\left|\boldsymbol{L}_h^{(3)}\right|}\left(I_3 - \frac{\boldsymbol{L}_h^{(3)}\boldsymbol{L}_h^{(3)^T}}{\left|\boldsymbol{L}_h^{(3)}\right|^2}\right) \\ \frac{1}{Z_h}I_3 & \frac{-\boldsymbol{Z}_0}{Z_h^2} & \frac{d_{\max}}{\left|\boldsymbol{L}_h^{(1)}\right|}\left(I_3 - \frac{\boldsymbol{L}_h^{(1)}\boldsymbol{L}_h^{(1)^T}}{\left|\boldsymbol{L}_h^{(1)}\right|^2}\right) & 0 \\ \frac{1}{Z_h}I_3 & \frac{-\boldsymbol{Z}_0}{Z_h^2} & 0 & \frac{d_{\max}}{\left|\boldsymbol{L}_h^{(3)}\right|}\left(I_3 - \frac{\boldsymbol{L}_h^{(3)}\boldsymbol{L}_h^{(3)^T}}{\left|\boldsymbol{L}_h^{(3)}\right|^2}\right) \end{pmatrix} \tag{2.263}$$

their covariance matrices are again obtained under the assumption that the projection center is uncorrelated with the projection ray directions, as

$$\begin{pmatrix} C_{X_1X_1} & C_{X_1X_2} & C_{X_1X_3} & C_{X_1X_4} \\ C_{X_2X_1} & C_{X_2X_2} & C_{X_2X_3} & C_{X_2X_4} \\ C_{X_3X_1} & C_{X_3X_2} & C_{X_3X_3} & C_{X_3X_4} \\ C_{X_4X_1} & C_{X_4X_2} & C_{X_4X_3} & C_{X_4X_4} \end{pmatrix} = J \begin{pmatrix} C_{ZZ} & 0 & 0 \\ 0 & C_{L_h^{(1)}L_h^{(1)}} & C_{L_h^{(1)}L_h^{(3)}} \\ 0 & C_{L_h^{(3)}L_h^{(1)}} & C_{L_h^{(3)}L_h^{(3)}} \end{pmatrix} J^T \tag{2.264}$$

The four uncertain oriented projective corner-points are easily obtained from those uncertain Euclidean points as described in section 2.1.3 and the resulting uncertain oriented projective 3d polygon (see figure 2.13) in point representation is given by

$$(\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4\}, \{C_{X_iX_j} | i, j = 1, ..., 4\}) \tag{2.265}$$

If two such polygons from two distinct cameras intersect in space, then the image line segments can be matched with each other. As the distances of the scene line from the cameras are taken into account, this allows to check the epipolar constraint for two line segments between an image pair. In the following section a method will be presented, how this intersection between two uncertain oriented projective space polygons can be tested.

Up to now the representation of entities in the uncertain oriented projective framework has been discussed. It has been shown, how to construct entities from other entities and how the uncertain oriented projective 2d space is related to the uncertain oriented projective 3d space by projective cameras, which are uncertain oriented projective 3d entities themselves. The use of oriented projective geometry instead of non-oriented projective geometry enabled the construction of line segments and polygons in 2d and 3d space. Furthermore, for each transformation it was explicitly shown, how all the uncertainties propagate. Those uncertainties will play a major role in the next section, where the task of geometric reasoning, i.e. making decisions about relations between uncertain oriented projective entities, will be described.

2.2 Testing geometric relations

The goal of representing the uncertainties of the geometric entities is to facilitate rigorous statistical testing, whether some geometric relation between the entities holds or not. In the following a notion of uncertain relations between the uncertain entities will be presented. Useful example relations between the base entities as well as the compound entities will be shown thereafter.

Usually a relation between to vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by some characteristic function

$$R(\boldsymbol{x}, \boldsymbol{y}) : I\!R^n \times I\!R^m \mapsto \{0, 1\} \tag{2.266}$$

indicating, whether the relation between the two entities holds or not. As discussed in the previous section, all uncertain entities are given by probability density functions $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$. A relation between two such uncertain entities is now assumed to hold, if the joint probability mass satisfying the relation is large enough, i.e.

$$\int_{\{(\boldsymbol{x},\boldsymbol{y}) \in I\!R^n \times I\!R^m | R(\boldsymbol{x},\boldsymbol{y})=1\}} p(\boldsymbol{x}) p(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} > \alpha \tag{2.267}$$

Observe that there is no assertion that the relation actually holds. However, in the absence of contradicting evidence, following an empiristic rather than rationalistic tradition, it will be treated as if in all of the following, meaning that an uncertain relation is considered to hold if and only if condition (2.267) holds. This is a much stronger notion of uncertain relations, than the notion resulting from statistical hypothesis testing, as non-falsified hypothesis are conjectured to be true.

Also note that, if the relation respects orientated homogeneity, i.e. if

$$R(\boldsymbol{x}, \boldsymbol{y}) = R(\lambda \boldsymbol{x}, \mu \boldsymbol{y}) \qquad \forall \lambda > 0, \mu > 0 \tag{2.268}$$

there is no need to distinguish between uncertain Euclidean and uncertain oriented projective entities. All relations, which are discussed here, have this property.

For full-rank bilinear relations, i.e. relations of the form

$$R(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 1 & \text{if } \mathsf{A}(\boldsymbol{x})\boldsymbol{y} = \mathsf{B}(\boldsymbol{y})\boldsymbol{x} = \boldsymbol{0} \\ 0 & \text{otherwise} \end{cases} \tag{2.269}$$

with $\mathsf{A}(\boldsymbol{x})$ and $\mathsf{B}(\boldsymbol{y})$ having full rank r, and Normal distributed uncertain entities $(\boldsymbol{x}, \mathsf{C}_{xx})$ and $(\boldsymbol{y}, \mathsf{C}_{yy})$ (which is justifiable from the maximum entropy assumption as discussed in

section 2.1) this test may be performed by checking (cf. [Heuel, 2004, p.130] and [Koch, 1997, p.134]), if the Mahalanobis-distance

$$\boldsymbol{y}^T \mathsf{A}(\boldsymbol{x})^T \left(\mathsf{A}(\boldsymbol{x}) \mathsf{C}_{yy} \mathsf{A}(\boldsymbol{x})^T + \mathsf{B}(\boldsymbol{y}) \mathsf{C}_{xx} \mathsf{B}(\boldsymbol{y})^T \right)^{-1} \mathsf{A}(\boldsymbol{x}) \boldsymbol{y} \leq T_{\alpha,r} \tag{2.270}$$

is smaller than a threshold, which is numerically derived as the α-fractile from the χ^2_r-distribution with r degrees of freedom

$$\frac{1}{2^{r/2} \int_0^\infty t^{r/2-1} e^{-t} dt} \int_0^{T_{\alpha,r}} v^{r/2-1} e^{-v/2} dv = \alpha \tag{2.271}$$

If the rank r of $\mathsf{A}(\boldsymbol{x})$ and $\mathsf{B}(\boldsymbol{y})$ is smaller than the number of rows, then either a pseudo-inverse has to be used in the test, or rows have to be deleted from both matrices in order to obtain full rank matrices. The proposed method is to choose those r rows, where the product of the L_∞-norm is maximal. The reduced matrices will be denoted with $(\mathsf{A}(\boldsymbol{x}))^{[r]}$ and $(\mathsf{B}(\boldsymbol{y}))^{[r]}$ in the following. This approach slightly deviates from the approach proposed in [Heuel, 2004, p.67ff], where the product of the L_2-norms was proposed.

In an orientation preserving framework, this test can be extended for scalar bi-linear relations to not only test, if the bi-linear expression is equal to zero, but also to check, if its sign is negative. That means relations of the form

$$R(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 1 & \text{if } \boldsymbol{a}(\boldsymbol{x}) \boldsymbol{y} = \boldsymbol{b}(\boldsymbol{y}) \boldsymbol{x} \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.272}$$

with $\boldsymbol{a}(\boldsymbol{x})$ and $\boldsymbol{b}(\boldsymbol{y})$ being vectors can be tested by taking the square root of condition (2.270) yielding the modified equivalent condition

$$\frac{\boldsymbol{a}(\boldsymbol{x}) \boldsymbol{y}}{\sqrt{\boldsymbol{a}(\boldsymbol{x}) \mathsf{C}_{yy} \boldsymbol{a}(\boldsymbol{x})^T + \boldsymbol{b}(\boldsymbol{y}) \mathsf{C}_{xx} \boldsymbol{b}(\boldsymbol{y})^T}} \leq \sqrt{T_{\alpha,1}} \tag{2.273}$$

for the relation to hold.

As pointed out before, holding uncertain relations are considered as being conjectures rather than non-falsified hypothesis. This allows to reason with uncertain entities using uncertain relations and Boolean operations. In the following a conjunction will be denoted as

$$\begin{aligned} R(\boldsymbol{x}, \boldsymbol{y}) &= R_1(\boldsymbol{x}, \boldsymbol{y}) \wedge R_2(\boldsymbol{x}, \boldsymbol{y}) && (2.274) \\ &= \min(R_1(\boldsymbol{x}, \boldsymbol{y}), R_2(\boldsymbol{x}, \boldsymbol{y})) && (2.275) \end{aligned}$$

and a disjunction will be denoted as

$$\begin{aligned} R(\boldsymbol{x}, \boldsymbol{y}) &= R_1(\boldsymbol{x}, \boldsymbol{y}) \vee R_2(\boldsymbol{x}, \boldsymbol{y}) && (2.276) \\ &= \max(R_1(\boldsymbol{x}, \boldsymbol{y}), R_2(\boldsymbol{x}, \boldsymbol{y})) && (2.277) \end{aligned}$$

Observe that the condition for the conjunction of the two bi-linear sign tests

$$\boldsymbol{a}(\boldsymbol{x}) \boldsymbol{y} \stackrel{!}{\leq} 0 \wedge -\boldsymbol{a}(\boldsymbol{x}) \boldsymbol{y} \stackrel{!}{\leq} 0 \tag{2.278}$$

is equivalent to the condition for the bi-linear test

$$\boldsymbol{a}(\boldsymbol{x}) \boldsymbol{y} \stackrel{!}{=} 0 \tag{2.279}$$

as would be expected.

In the following some example relations for the entities presented in the previous section will be shown.

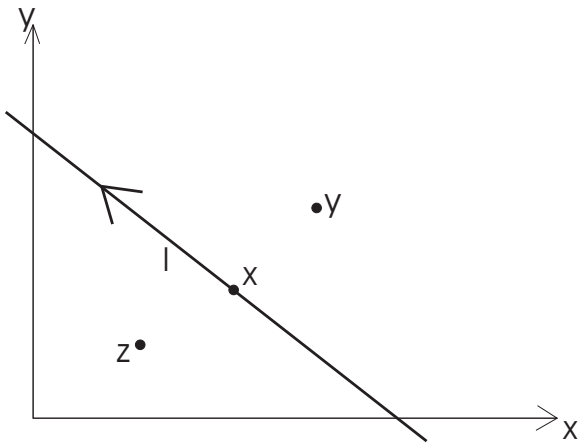

Figure 2.14: The 2d point $\mathbf{x}$ is incident to the 2d line $\mathbf{l}$. The 2d point $\mathbf{y}$ is right of $\mathbf{l}$ and the 2d point $\mathbf{z}$ is left of $\mathbf{l}$.

2.2.1 Relations between the base entities

In the previous section two special types of relations together with a method for checking those relations for uncertain oriented projective entities were presented. In the following a range of such relations including incidence, equality, orthogonality and parallelity between the base entities will be presented and for each of those relations the required Jacobians will be made visible.

Incidence

To check, whether an oriented projective 2d point $\mathbf{x}$ is incident to an oriented projective 2d line $\mathbf{l}$, the inner product must be zero (see figure 2.14). This will be denoted as

$$\mathsf{Incident}(\mathbf{x},\mathbf{l}) \Leftrightarrow \mathbf{x}^T\mathbf{l} = \mathbf{l}^T\mathbf{x} \stackrel{!}{=} 0 \tag{2.280}$$

Observe that the covariance matrices are only left out in this notation in order to improve the readability. They are still required, though, and must be present for every entity as has been discussed in section 2.1.

To see if the oriented projective 2d point $\mathbf{x}$ is left or right of the oriented projective 2d line $\mathbf{l}$, one has to check the sign of the scalar product (see figure 2.14), therefore the relations are given by

$$\mathsf{Right}(\mathbf{x},\mathbf{l}) \Leftrightarrow \mathbf{x}^T\mathbf{l} = \mathbf{l}^T\mathbf{x} \stackrel{!}{\leq} 0 \tag{2.281}$$

and

$$\mathsf{Left}(\mathbf{x},\mathbf{l}) \Leftrightarrow -\mathbf{x}^T\mathbf{l} = -\mathbf{l}^T\mathbf{x} \stackrel{!}{\leq} 0 \tag{2.282}$$

In 3d the situation between oriented 3d points $\mathbf{X}$ and oriented 3d planes $\mathbf{A}$ is just the same. The two entities are incident, if their scalar product is equal to zero (see figure 2.15), hence one defines the relation

$$\mathsf{Incident}(\mathbf{X},\mathbf{A}) \Leftrightarrow \mathbf{X}^T\mathbf{A} = \mathbf{A}^T\mathbf{X} \stackrel{!}{=} 0 \tag{2.283}$$

A similar directed notion is possible, too, and one defines an oriented projective 3d point $\mathbf{X}$ to be above or below of an oriented projective 3d plane $\mathbf{A}$ as (see figure 2.15)

$$\mathsf{Below}(\mathbf{X},\mathbf{A}) \Leftrightarrow \mathbf{X}^T\mathbf{A} = \mathbf{A}^T\mathbf{X} \stackrel{!}{\leq} 0 \tag{2.284}$$

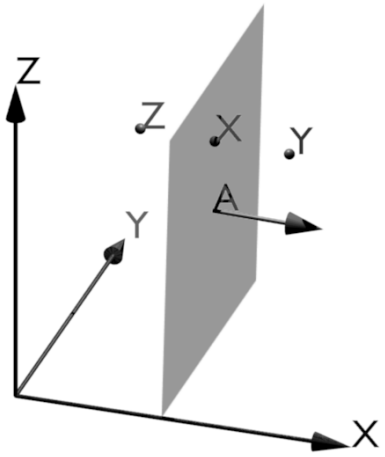

Figure 2.15: The 3d point $\mathbf{X}$ is incident to the 3d plane $\mathbf{A}$. The 3d point $\mathbf{Y}$ is above the 3d plane $\mathbf{A}$ and the 3d point $\mathbf{Z}$ is below the 3d plane $\mathbf{A}$.

and

$$\mathsf{Above}(\mathbf{X},\mathbf{A}) \Leftrightarrow -\mathbf{X}^T\mathbf{A} = -\mathbf{A}^T\mathbf{X} \stackrel{!}{\leq} 0 \tag{2.285}$$

Two projective 3d lines $\mathbf{L}$ and $\mathbf{M}$ intersect, if their inner product is equal to zero (see figure 2.16). Therefore one defines

$$\mathsf{Incident}(\mathbf{L},\mathbf{M}) \Leftrightarrow \mathbf{L}^T D_6 \mathbf{M} = \mathbf{M}^T D_6 \mathbf{L} \stackrel{!}{=} 0 \tag{2.286}$$

In a similar fashion the two oriented versions (see figure 2.16)

$$\mathsf{Left}(\mathbf{L},\mathbf{M}) \Leftrightarrow \mathbf{L}^T D_6 \mathbf{M} = \mathbf{M}^T D_6 \mathbf{L} \stackrel{!}{\leq} 0 \tag{2.287}$$

and

$$\mathsf{Right}(\mathbf{L},\mathbf{M}) \Leftrightarrow -\mathbf{L}^T D_6 \mathbf{M} = -\mathbf{M}^T D_6 \mathbf{L} \stackrel{!}{\leq} 0 \tag{2.288}$$

may be defined. The two relations describe the screw between the two oriented 3d lines (cf. [Förstner and Wrobel, 2004, p.139]), which means one line can be transformed on a shortest path into the other line by pushing it and simultaneously rotating it either to the left or to the right.

A projective 3d point $\mathbf{X}$ lies on a projective 3d line $\mathbf{L}$, if the 3d plane constructed from both entities does not exist (see figure 2.17), i.e. (cf. (2.95))

$$\overline{\mathbf{\Pi}}(\mathbf{X})^T\mathbf{L} = \overline{\mathbf{\Gamma}}(\mathbf{L})^T\mathbf{X} \stackrel{!}{=} \mathbf{0} \tag{2.289}$$

This expression is bi-linear, however it has not full rank. To facilitate checking uncertain relations as described above, one has to consistently reduce the two matrices $\overline{\mathbf{\Pi}}(\mathbf{X})^T$ and $\overline{\mathbf{\Gamma}}(\mathbf{L})^T$, so that they have the full rank of two (cf. [Heuel, 2004, p.67]). Hence incidence is defined as

$$\mathsf{Incident}(\mathbf{X},\mathbf{L}) \Leftrightarrow \left(\overline{\mathbf{\Pi}}(\mathbf{X})^T\right)^{[2]}\mathbf{L} = \left(\overline{\mathbf{\Gamma}}(\mathbf{L})^T)\right)^{[2]}\mathbf{X} \stackrel{!}{=} \mathbf{0} \tag{2.290}$$

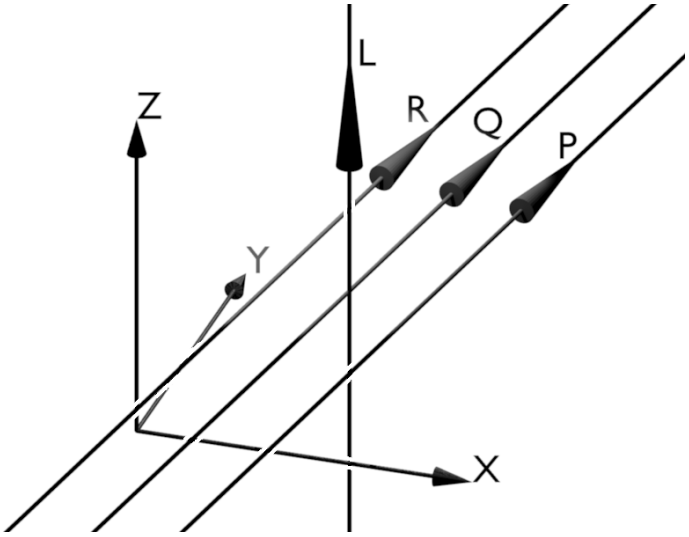

Figure 2.16: The 3d lines **Q** and **L** are incident. The 3d lines **R** and **L** are right-screwed and the 3d lines **P** and **L** are left-screwed.

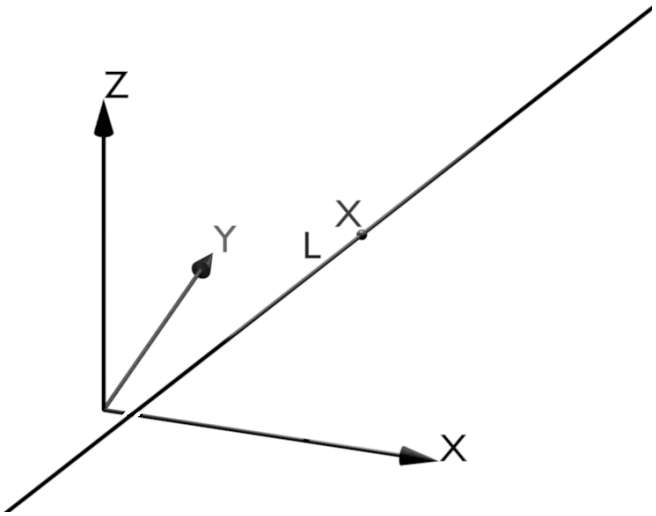

Figure 2.17: The 3d point **X** is incident to the 3d line **L**. In this case no direct notion of left or right exists, as the incidence relation between a 3d point and a 3d line has two degrees of freedom.

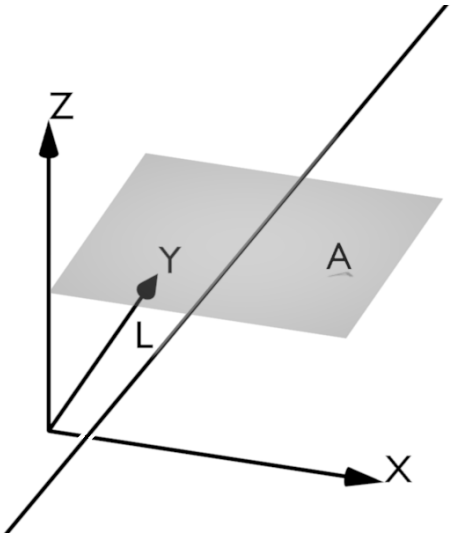

Figure 2.18: The 3d line $\mathbf{L}$ is incident to the 3d plane $\mathbf{A}$. Note that no notion of $\mathbf{L}$ being above or below $\mathbf{A}$ can be defined unless $\mathbf{L}$ and $\mathbf{A}$ are parallel.

For duality reasons the incidence relation of a projective 3d line $\mathbf{L}$ with a projective 3d plane $\mathbf{A}$ (see figure 2.18) is defined as (cf. (2.106))

$$\mathsf{Incident}(\mathbf{A},\mathbf{L}) \Leftrightarrow \left(\mathbf{\Pi}(\mathbf{A})^T\right)^{[2]}\mathbf{L} = \left(\mathbf{\Gamma}(\mathbf{L})^T)\right)^{[2]}\mathbf{A} \stackrel{!}{=} \mathbf{0} \tag{2.291}$$

Identity

Two oriented projective 2d points $\mathbf{x}$ and $\mathbf{y}$ are considered equal up to orientation, if it is not possible, to construct a joining line. Therefore the equality relation up to orientation is defined as (cf. (2.32))

$$\mathsf{Equal}(\mathbf{x},\mathbf{y}) \Leftrightarrow (\mathcal{S}(\mathbf{x}))^{[2]}\mathbf{y} = -\left(\mathcal{S}(\mathbf{y})\right))^{[2]}\mathbf{x} \stackrel{!}{=} \mathbf{0} \tag{2.292}$$

The orientation cannot be checked in a bi-linear test and can easily be tested in an extra step if required. For duality reasons the equality of two projective 2d lines $\mathbf{l}$ and $\mathbf{m}$ up to orientation is defined as (cf. (2.38))

$$\mathsf{Equal}(\mathbf{l},\mathbf{m}) \Leftrightarrow (\mathcal{S}(\mathbf{l}))^{[2]}\mathbf{m} = -\left(\mathcal{S}(\mathbf{m})\right))^{[2]}\mathbf{l} \stackrel{!}{=} \mathbf{0} \tag{2.293}$$

In 3d an analogous argument applies. Two projective 3d points $\mathbf{X}$ and $\mathbf{Y}$ are considered to be equal up to orientation, if it is not possible to construct a joining line. Hence, equality up to orientation for projective 3d points is defined as (cf. (2.83))

$$\mathsf{Equal}(\mathbf{X},\mathbf{Y}) \Leftrightarrow (\mathbf{\Pi}(\mathbf{X}))^{[3]}\mathbf{Y} = -\left(\mathbf{\Pi}(\mathbf{Y})\right))^{[3]}\mathbf{X} \stackrel{!}{=} \mathbf{0} \tag{2.294}$$

For duality reasons this also applies to projective 3d planes $\mathbf{A}$ and $\mathbf{B}$, where equality is defined as (cf. (2.109))

$$\mathsf{Equal}(\mathbf{A},\mathbf{B}) \Leftrightarrow \left(\overline{\mathbf{\Pi}}(\mathbf{A})\right)^{[3]}\mathbf{B} = -\left(\overline{\mathbf{\Pi}}(\mathbf{B}))\right)^{[3]}\mathbf{A} \stackrel{!}{=} \mathbf{0} \tag{2.295}$$

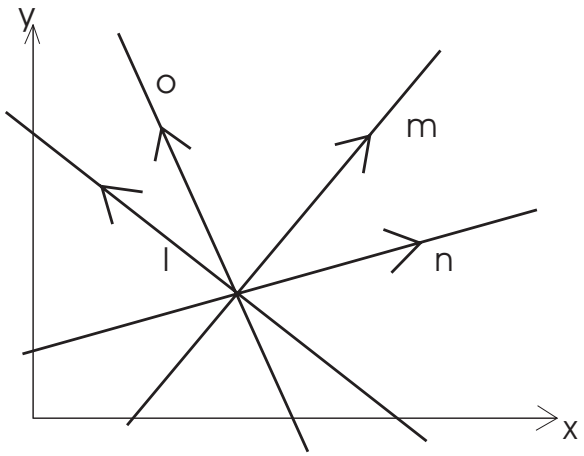

Figure 2.19: The 2d lines $\mathbf{m}$ and $\mathbf{l}$ are orthogonal. The directed 2d lines $\mathbf{o}$ and $\mathbf{l}$ are concurrent and the directed 2d lines $\mathbf{n}$ and $\mathbf{l}$ are countercurrent.

Defining equality for 3d lines is a little more involved. As pointed out on page 74ff in [Heuel, 2004], it is in principle possible to formulate as bi-linear relation, but the required reduction is not trivial. The proposed solution (cf. [Heuel, 2004, p.76]) is as follows: Given two projective 3d lines $\mathbf{L}$ and $\mathbf{M}$ an index $i \in \{1, ..., 6\}$ is selected, where the product $|L_i M_i| >> 0$ is maximal. If no such index exists, the lines are not possibly identical. Now equality is defined as

$$\mathsf{Equal}(\mathbf{L}, \mathbf{M}) \Leftrightarrow (\boldsymbol{\Delta}_i(\mathbf{L}))^{[4]}\, \mathbf{M} = -\,(\boldsymbol{\Delta}_i(\mathbf{M}))^{[4]}\, \mathbf{L} \stackrel{!}{=} \mathbf{0} \tag{2.296}$$

with

$$\boldsymbol{\Delta}_i(\mathbf{L}) = \mathbf{L}\boldsymbol{E}_i^T - L_i I_6 \tag{2.297}$$

where $\boldsymbol{E}_i$ denotes the 6-vector, containing zeros except for position i, where it contains a one.

Orthogonality

As already used in equation (2.52) the orthogonality of two projective 2d lines $\mathbf{l}$ and $\mathbf{m}$ can be defined using the conic C_∞^* (cf. equation (2.51)), which is dual to the two circular points (cf. [Hartley and Zisserman, 2000, p.33f]). Hence, orthogonality is defined as (see figure 2.19)

$$\mathsf{Orthogonal}(\mathbf{l}, \mathbf{m}) \Leftrightarrow \mathbf{l}^T \mathsf{C}_\infty^* \mathbf{m} = \mathbf{m}^T \mathsf{C}_\infty^* \mathbf{l} \stackrel{!}{=} 0 \tag{2.298}$$

The oriented version of this relation indicates, if the two oriented lines point into opposite directions (see figure 2.19)

$$\mathsf{Countercurrent}(\mathbf{l}, \mathbf{m}) \Leftrightarrow \mathbf{l}^T \mathsf{C}_\infty^* \mathbf{m} = \mathbf{m}^T \mathsf{C}_\infty^* \mathbf{l} \stackrel{!}{\leq} 0 \tag{2.299}$$

or if they point into the same direction (see figure 2.19)

$$\mathsf{Concurrent}(\mathbf{l}, \mathbf{m}) \Leftrightarrow -\mathbf{l}^T \mathsf{C}_\infty^* \mathbf{m} = -\mathbf{m}^T \mathsf{C}_\infty^* \mathbf{l} \stackrel{!}{\leq} 0 \tag{2.300}$$

The same argument is possible for 3d planes, as already discussed in equation (2.149). Therefore orthogonality is defined for projective 3d planes $\mathbf{A}$ and $\mathbf{B}$ using the conic $\tilde{\mathsf{C}}_\infty^*$ (cf. (2.148)) as (see figure 2.20)

$$\mathsf{Orthogonal}(\mathbf{A}, \mathbf{B}) \Leftrightarrow \mathbf{A}^T \tilde{\mathsf{C}}_\infty^* \mathbf{B} = \mathbf{B}^T \tilde{\mathsf{C}}_\infty^* \mathbf{A} \stackrel{!}{=} 0 \tag{2.301}$$

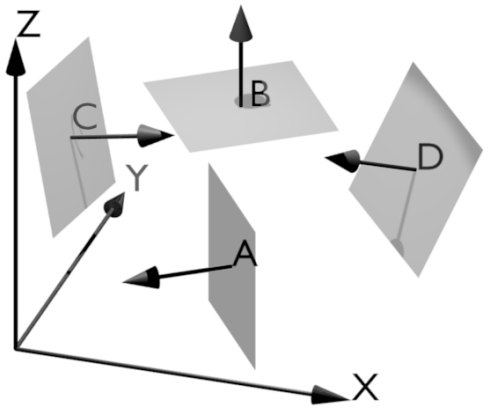

Figure 2.20: The two 3d planes $\mathbf{A}$ and $\mathbf{B}$ are orthogonal. The two 3d planes $\mathbf{A}$ and $\mathbf{C}$ face into opposite directions and the two 3d planes $\mathbf{A}$ and $\mathbf{D}$ face into the same direction.

Again an oriented version is possible. The two oriented planes face opposite directions (see figure 2.20), if

$$\mathsf{OppositeFacing}(\mathbf{A},\mathbf{B}) \Leftrightarrow \mathbf{A}^T \tilde{C}_\infty^* \mathbf{B} = \mathbf{B}^T \tilde{C}_\infty^* \mathbf{A} \stackrel{!}{\leq} 0 \tag{2.302}$$

and face the same direction (see figure 2.20), if

$$\mathsf{SameFacing}(\mathbf{A},\mathbf{B}) \Leftrightarrow -\mathbf{A}^T \tilde{C}_\infty^* \mathbf{B} = -\mathbf{B}^T \tilde{C}_\infty^* \mathbf{A} \stackrel{!}{\leq} 0 \tag{2.303}$$

This definition for projective 3d planes is easily extended for projective 3d lines. The orthogonality of two projective 3d lines $\mathbf{L}$ and $\mathbf{M}$ is defined as (see figure 2.21)

$$\mathsf{Orthogonal}(\mathbf{L},\mathbf{M}) \Leftrightarrow \mathbf{L}^T \begin{pmatrix} I_3 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{M} = \mathbf{M}^T \begin{pmatrix} I_3 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{L} \stackrel{!}{=} 0 \tag{2.304}$$

As in the 2d case, the oriented versions indicate, whether the two oriented lines point into opposite directions (see figure 2.21)

$$\mathsf{Countercurrent}(\mathbf{L},\mathbf{M}) \Leftrightarrow \mathbf{L}^T \begin{pmatrix} I_3 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{M} = \mathbf{M}^T \begin{pmatrix} I_3 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{L} \stackrel{!}{\leq} 0 \tag{2.305}$$

or if they point into the same direction (see figure 2.21)

$$\mathsf{Concurrent}(\mathbf{L},\mathbf{M}) \Leftrightarrow -\mathbf{L}^T \begin{pmatrix} I_3 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{M} = -\mathbf{M}^T \begin{pmatrix} I_3 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{L} \stackrel{!}{\leq} 0 \tag{2.306}$$

Finally the orthogonality of a projective 3d plane and a projective 3d line has already been used in equation (2.126). Those two are considered orthogonal, if the direction vector of the line and the normal vector of the plane point into the same direction (see figure 2.22). This can be either expressed using the extended $\tilde{\epsilon}_{rst}$ tensor (cf. (2.125)) or using the skew symmetric matrix. It is defined as

$$\mathsf{Orthogonal}(\mathbf{L},\mathbf{A}) \Leftrightarrow \begin{pmatrix} S(\boldsymbol{L}_h) & \mathbf{0} \end{pmatrix}^{[2]} \mathbf{A} = \begin{pmatrix} -S(\boldsymbol{A}_h) & 0 \end{pmatrix}^{[2]} \mathbf{L} \stackrel{!}{=} \mathbf{0} \tag{2.307}$$

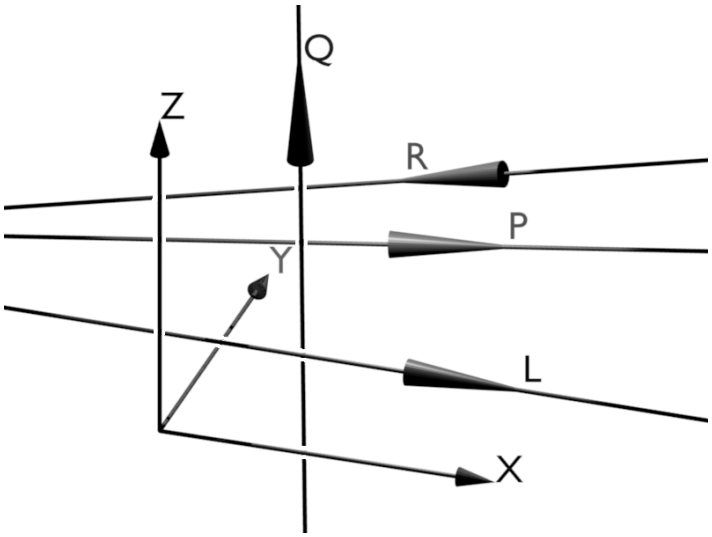

Figure 2.21: The two 3d lines **L** and **Q** are orthogonal. The two 3d lines **L** and **P** are concurrent and the two 3d lines **L** and **R** are countercurrent.

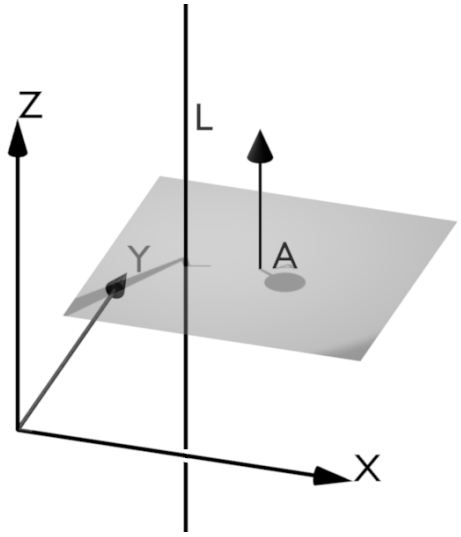

Figure 2.22: The 3d line **L** and the 3d plane **A** are orthogonal. Note that no analogous oriented notion is definable in this case.

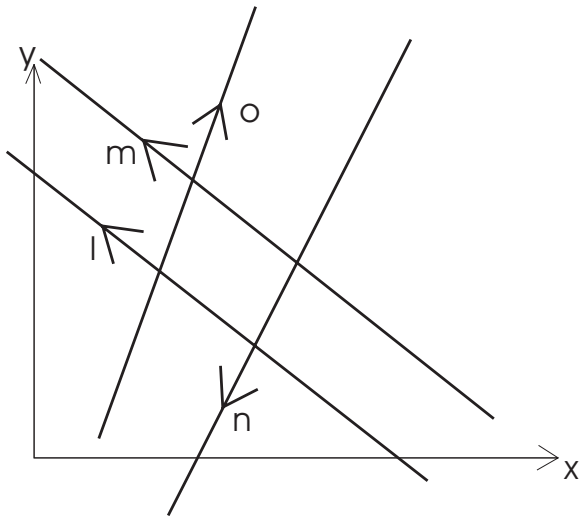

Figure 2.23: The two 2d lines $\mathbf{m}$ and $\mathbf{l}$ are parallel. The 2d line $\mathbf{o}$ crosses $\mathbf{l}$ from left and the 2d line $\mathbf{n}$ crosses $\mathbf{l}$ from right.

Parallelity

Two projective 2d lines $\mathbf{l}$ and $\mathbf{m}$ are considered parallel, if one rotated by 90° is orthogonal to the other. Therefore using the rotation matrix (cf. (2.46))

$$R_\perp = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.308}$$

parallelity is defined as (see figure 2.23)

$$\mathsf{Parallel}(\mathbf{l},\mathbf{m}) \Leftrightarrow \mathbf{l}^T R_\perp^T C_\infty^* \mathbf{m} = \mathbf{m}^T C_\infty^* R_\perp \mathbf{l} \stackrel{!}{=} 0 \tag{2.309}$$

The oriented version of this relation indicates, whether the one of the lines crosses the other from the left (see figure 2.23)

$$\mathsf{FromLeft}(\mathbf{l},\mathbf{m}) \Leftrightarrow \mathbf{l}^T R_\perp^T C_\infty^* \mathbf{m} = \mathbf{m}^T C_\infty^* R_\perp \mathbf{l} \stackrel{!}{\leq} 0 \tag{2.310}$$

or from the right (see figure 2.23)

$$\mathsf{FromRight}(\mathbf{l},\mathbf{m}) \Leftrightarrow -\mathbf{l}^T R_\perp^T C_\infty^* \mathbf{m} = -\mathbf{m}^T C_\infty^* R_\perp \mathbf{l} \stackrel{!}{\leq} 0 \tag{2.311}$$

A projective 3d line $\mathbf{L}$ and a projective 3d plane $\mathbf{A}$ are considered parallel, if the direction of the line and the normal of the plane are orthogonal (see figure 2.24). Parallelity is therefore defined as

$$\mathsf{Parallel}(\mathbf{L},\mathbf{A}) \Leftrightarrow \mathbf{L}^T \begin{pmatrix} \tilde{C}_\infty^* \\ 0 \end{pmatrix} \mathbf{A} = \mathbf{A}^T \begin{pmatrix} \tilde{C}_\infty^* & 0 \end{pmatrix} \mathbf{L} \stackrel{!}{=} 0 \tag{2.312}$$

As in the 2d case, the oriented version of this relation indicates, whether the lines crosses the plane from above (see figure 2.24)

$$\mathsf{FromAbove}(\mathbf{L},\mathbf{A}) \Leftrightarrow \mathbf{L}^T \begin{pmatrix} \tilde{C}_\infty^* \\ 0 \end{pmatrix} \mathbf{A} = \mathbf{A}^T \begin{pmatrix} \tilde{C}_\infty^* & 0 \end{pmatrix} \mathbf{L} \stackrel{!}{\leq} 0 \tag{2.313}$$

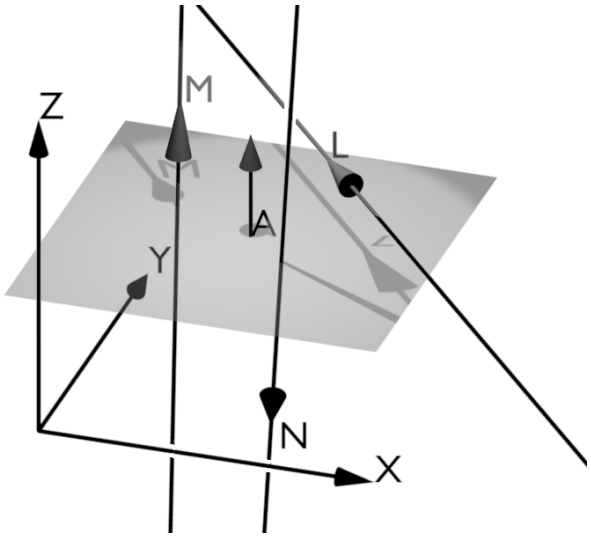

Figure 2.24: The 3d line **L** and the 3d plane **A** are parallel. The 3d line **N** crosses **A** from above, the 3d line **M** crosses **A** from below.

or from below (see figure 2.24)

$$\mathsf{FromBelow}(\mathbf{L},\mathbf{A}) \Leftrightarrow -\mathbf{L}^T \begin{pmatrix} \tilde{C}_\infty^* \\ 0 \end{pmatrix} \mathbf{A} = -\mathbf{A}^T \begin{pmatrix} \tilde{C}_\infty^* & 0 \end{pmatrix} \mathbf{L} \stackrel{!}{\leq} 0 \tag{2.314}$$

Two projective 3d planes are considered to be parallel, if their normals point into the same direction (see figure 2.25). It is therefore defined as

$$\mathsf{Parallel}(\mathbf{A},\mathbf{B}) \Leftrightarrow \begin{pmatrix} S(\boldsymbol{A}_h) & \mathbf{0} \end{pmatrix}^{[2]} \mathbf{B} = \begin{pmatrix} -S(\boldsymbol{B}_h) & \mathbf{0} \end{pmatrix}^{[2]} \mathbf{A} \stackrel{!}{=} \mathbf{0} \tag{2.315}$$

The same argument holds for two projective 3d lines **L** and **M**, which are considered to be parallel, if their direction vectors point in the same direction, so that parallelity is defined analogously as (see figure 2.26)

$$\mathsf{Parallel}(\mathbf{L},\mathbf{M}) \Leftrightarrow \begin{pmatrix} S(\boldsymbol{L}_h) & 0 \end{pmatrix}^{[2]} \mathbf{M} = \begin{pmatrix} -S(\boldsymbol{M}_h) & 0 \end{pmatrix}^{[2]} \mathbf{L} \stackrel{!}{=} \mathbf{0} \tag{2.316}$$

Up to now various relations between the base entities have been presented. Those relations were either bi-linear tests or bi-linear sign tests containing two base entities. No conjunctions or disjunctions have been considered so far. In the following conjunctions and disjunctions of the relations between the base entities presented so far will be used to derive useful relations between the compound entities.

2.2.2 Relations between the compound entities

In the previous section various relations between the base entities were presented. Those relations can of course be checked also for the components of the compound entities, such as line segments and polygons in 2d and 3d space. In addition to the basic tests, several more meaningful tests can be constructed from conjunctions and disjunctions of those. In the following each compound entity discussed in section 2.1 will be picked up in turn and compound relations with these entities will be derived.

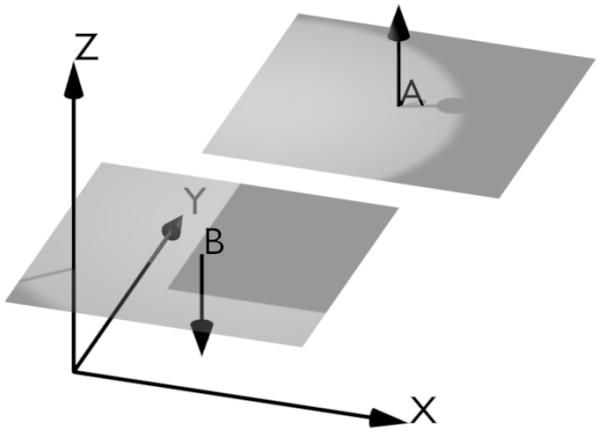

Figure 2.25: The two plane **A** and **B** are parallel. Note that no analogous oriented notion is definable in this case.

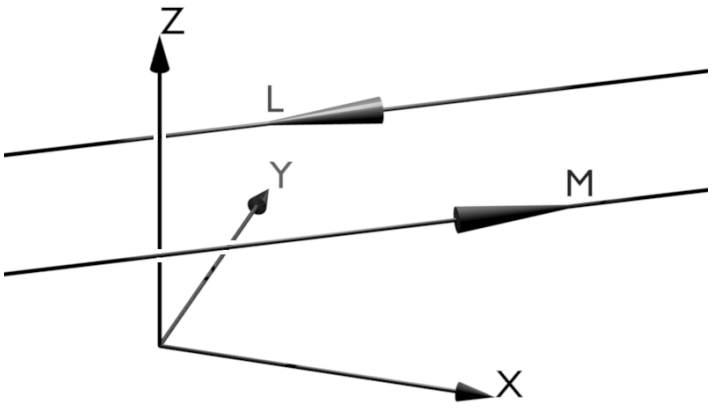

Figure 2.26: The two 3d lines **L** and **M** are parallel. Note that no analogous oriented notion is definable in this case.

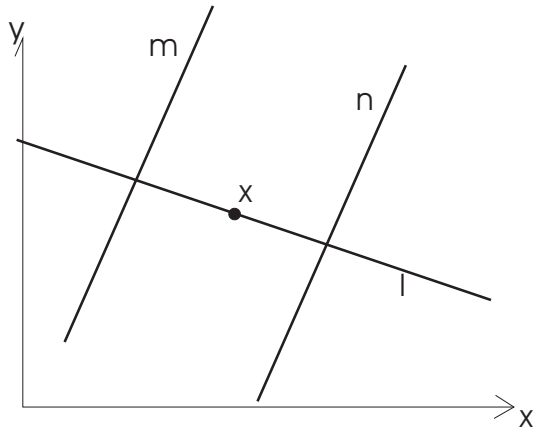

Figure 2.27: The 2d point $\mathbf{x}$ is incident to the 2d line segment, because it is incident to the 2d line $\mathbf{l}$ and between the two 2d lines $\mathbf{m}$ and $\mathbf{n}$.

2d line segments

Starting with the 2d line segment, the conjunctions and disjunctions that need to tested in order to decide on the incidence of a 2d point to a 2d line segment, the incidence of two 2d line segments, the intersection of two 2d line segments as well as the orthogonality and parallelity of two 2d line segments will be shown.

Incidence: Given an uncertain oriented 2d line segment in line representation

$$s : (\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}, \{\mathsf{C}_{ll}, \mathsf{C}_{mm}, \mathsf{C}_{nn}\}) \tag{2.317}$$

an uncertain oriented 2d point $(\mathbf{x}, \mathsf{C}_{xx})$ lies on this line segment, if it is on the line $\mathbf{l}$ and between the lines $\mathbf{m}$ and $\mathbf{n}$ (see figure 2.27). First observe that a point lies between the two delimiting lines, if

$$\mathsf{Between}(\mathbf{x}, \mathbf{m}, \mathbf{n}) \Leftrightarrow \mathsf{Right}(\mathbf{x}, \mathbf{m}) \wedge \mathsf{Right}(\mathbf{x}, \mathbf{n}) \tag{2.318}$$

Now incidence of a point and a line segment is defined as the condition

$$\begin{aligned} &\mathsf{Incident}(\mathbf{x}, s) \\ &\quad \Leftrightarrow \mathsf{Incident}(\mathbf{x}, \mathbf{l}) \wedge \mathsf{Between}(\mathbf{x}, \mathbf{m}, \mathbf{n}) \end{aligned} \tag{2.319}$$

which holds, whenever there is no reason to assume that the point does not lie on the line segment.

Two uncertain oriented 2d line segments in line representation as well as in point representation

$$\begin{aligned} s_1 : (\{\mathbf{l}_1, \mathbf{m}_1, \mathbf{n}_1\}, \{\mathsf{C}_{l_1 l_1}, \mathsf{C}_{m_1 m_1}, \mathsf{C}_{n_1 n_1}\}) \\ \leftrightarrow (\{\mathbf{x}_1, \mathbf{y}_1\}, \{\mathsf{C}_{x_1 x_1}, \mathsf{C}_{y_1 y_1}\}) \end{aligned} \tag{2.320}$$

and

$$\begin{aligned} s_2 : (\{\mathbf{l}_2, \mathbf{m}_2, \mathbf{n}_2\}, \{\mathsf{C}_{l_2 l_2}, \mathsf{C}_{m_2 m_2}, \mathsf{C}_{n_2 n_2}\}) \\ \leftrightarrow (\{\mathbf{x}_2, \mathbf{y}_2\}, \{\mathsf{C}_{x_2 x_2}, \mathsf{C}_{y_2 y_2}\}) \end{aligned} \tag{2.321}$$

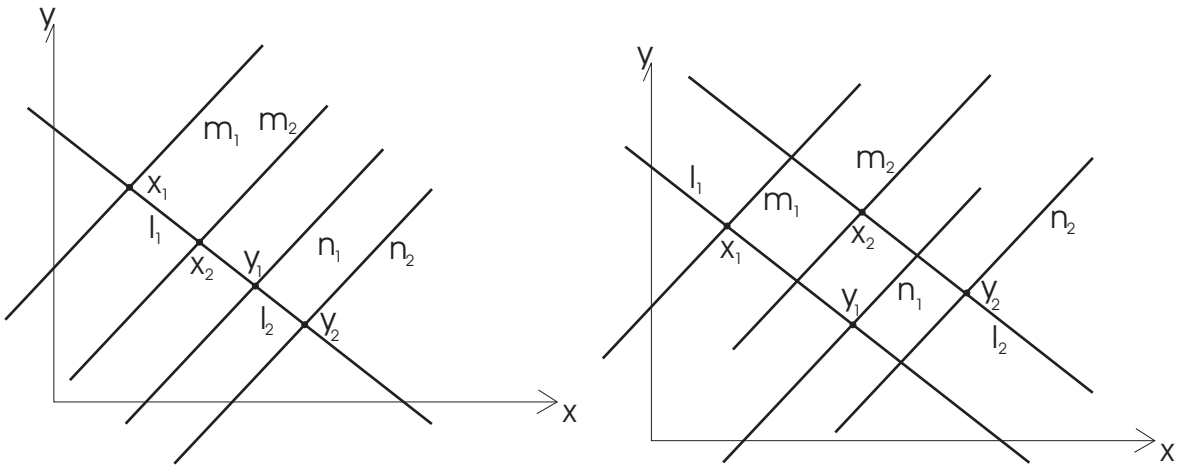

Figure 2.28: *Left:* The two depicted 2d line segments overlap because the two lines $\mathbf{l}_1$ and $\mathbf{l}_2$ are equal and the 2d point $\mathbf{x}_2$ is between the two 2d lines $\mathbf{m}_1$ and $\mathbf{n}_1$ and the 2d point $\mathbf{x}_1$ is between the two 2d lines $\mathbf{m}_2$ and $\mathbf{n}_2$. *Right:* The two 2d line segments are parallel because the 2d point $\mathbf{x}_2$ is between the two 2d lines $\mathbf{m}_1$ and $\mathbf{n}_1$ and the 2d point $\mathbf{x}_1$ is between the two 2d lines $\mathbf{m}_2$ and $\mathbf{n}_2$.

are considered to overlap, if the two connecting lines are equal and at least one end-point lies on the other segment (see figure 2.28, left). Hence overlap for two line segments is defined as

$$\begin{aligned}&\mathsf{Overlap}(\mathcal{s}_1, \mathcal{s}_2) \\ &\quad \Leftrightarrow \mathsf{Equal}(\mathbf{l}_1, \mathbf{l}_2) \wedge (\mathsf{Between}(\mathbf{x}_1, \mathbf{m}_2, \mathbf{n}_2) \vee \mathsf{Between}(\mathbf{y}_1, \mathbf{m}_2, \mathbf{n}_2) \vee \\ &\qquad \mathsf{Between}(\mathbf{x}_2, \mathbf{m}_1, \mathbf{n}_1) \vee \mathsf{Between}(\mathbf{y}_2, \mathbf{m}_1, \mathbf{n}_1))\end{aligned} \tag{2.322}$$

Parallelity: In contrast to infinite lines two line segments are considered to be parallel, if in addition to the parallelity of the connecting line, the orthogonal projection of at least one end-point on the other line lies on the segment (see figure 2.28, right). Hence, parallelity is defined for two line segments very analogous to incidence as

$$\begin{aligned}&\mathsf{Parallel}(\mathcal{s}_1, \mathcal{s}_2) \\ &\quad \Leftrightarrow \mathsf{Parallel}(\mathbf{l}_1, \mathbf{l}_2) \wedge (\mathsf{Between}(\mathbf{x}_1, \mathbf{m}_2, \mathbf{n}_2) \vee \mathsf{Between}(\mathbf{y}_1, \mathbf{m}_2, \mathbf{n}_2) \vee \\ &\qquad \mathsf{Between}(\mathbf{x}_2, \mathbf{m}_1, \mathbf{n}_1) \vee \mathsf{Between}(\mathbf{y}_2, \mathbf{m}_1, \mathbf{n}_1))\end{aligned} \tag{2.323}$$

Intersection: A line segment intersects a line, if its end-points lie on opposite sides of the line (see figure 2.29, left). Introducing for further reference the abbreviation

$$\begin{aligned}\mathsf{Opposite}(\mathbf{x}, \mathbf{y}, \mathbf{l}) \quad \Leftrightarrow \quad &(\mathsf{Left}(\mathbf{x}, \mathbf{l}) \wedge \mathsf{Right}(\mathbf{y}, \mathbf{l})) \vee \\ &(\mathsf{Right}(\mathbf{x}, \mathbf{l}) \wedge \mathsf{Left}(\mathbf{y}, \mathbf{l}))\end{aligned} \tag{2.324}$$

there is no reason to assume that a line segment does not intersect a line $\mathbf{l}$, if the following condition holds

$$\mathsf{Intersect}(\mathcal{s}_1, \mathbf{l}) \Leftrightarrow \mathsf{Opposite}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{l}) \tag{2.325}$$

Two line segments intersect, if the end-points of each segment lie on opposite sides of the joining line of the other (see figure 2.29, left). Hence, there is no reason to assume that the two line segments do not intersect, if the following condition holds

$$\mathsf{Intersect}(\mathcal{s}_1, \mathcal{s}_2) \Leftrightarrow \mathsf{Opposite}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{l}_2) \wedge \mathsf{Opposite}(\mathbf{x}_2, \mathbf{y}_2, \mathbf{l}_1) \tag{2.326}$$

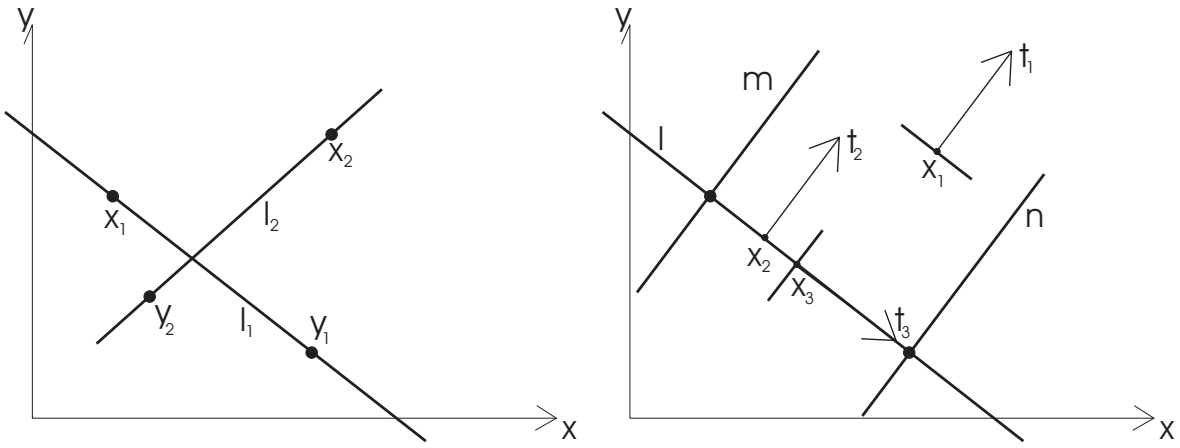

Figure 2.29: *Left:* The 2d line segment $\{\mathbf{x}_1, \mathbf{y}_1\}$ intersect the line $\mathbf{l}_2$, because the two 2d points $\mathbf{x}_1$ and $\mathbf{y}_1$ are on opposite sides of the 2d line $\mathbf{l}_2$. The two depicted line segments intersect, because also the 2d points the 2d points x_2 and $\mathbf{y}_2$ are on opposite sides of the 2d line $\mathbf{l}_1$. *Right:* The oriented projective 2d edgel $\{\mathbf{x}_1, \mathbf{t}_1\}$ is parallel to the 2d line $\mathbf{l}$. In addition to this, it is between the 2d lines $\mathbf{m}$ and $\mathbf{n}$, so that it is also considered parallel to the 2d line segment $\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}$. The oriented projective 2d edgel $\{\mathbf{x}_2, \mathbf{t}_2\}$ is incident to the 2d line $\mathbf{l}$. In addition to this, it is between the 2d lines $\mathbf{m}$ and $\mathbf{n}$, so that it is also considered incident to the 2d line segment $\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}$. Finally, the oriented projective 2d edgel $\{\mathbf{x}_3, \mathbf{t}_3\}$ is orthogonal to the 2d line $\mathbf{l}$. In addition to this, it is between the 2d lines $\mathbf{m}$ and $\mathbf{n}$, so that it is also considered orthogonal to the 2d line segment $\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}$.

Orthogonality: The previous definition is easily expendable toward orthogonal intersection by just adding one aditional term as follows

$$\begin{aligned} &\mathsf{Orthogonal}(\mathcal{s}_1, \mathcal{s}_2) \\ &\quad \Leftrightarrow \mathsf{Orthogonal}(\mathbf{l}_1, \mathbf{l}_2) \wedge \mathsf{Opposite}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{l}_2) \wedge \mathsf{Opposite}(\mathbf{x}_2, \mathbf{y}_2, \mathbf{l}_1) \end{aligned} \tag{2.327}$$

Identity: Two line segments are considered to be identical, if their end-points are identical. Because a line segment has a direction, i.e. a start- and an end-point, equality is defined as

$$\mathsf{Equal}(\mathcal{s}_1, \mathcal{s}_2) \Leftrightarrow \mathsf{Equal}(\mathbf{x}_1, \mathbf{x}_2) \wedge \mathsf{Equal}(\mathbf{y}_1, \mathbf{y}_2) \tag{2.328}$$

Note that the equality relation for the base entities is up to sign, so that a line segment passing the line at infinity can be considered equal to a finite line segment as long as their end-points are identical up to orientation.

2d edgels

Now the relations containing 2d edgels will be discussed. An uncertain oriented projective 2d edgel

$$e : (\{\mathbf{x}, \mathbf{t}\}, \{\mathsf{C}_{xx}, \mathsf{C}_{tt}\}) \tag{2.329}$$

is incident to an uncertain oriented projective 2d line $(\mathbf{l}, \mathsf{C}_{ll})$, if the point $\mathbf{x}$ is incident to the line $\mathbf{l}$ and the normal of the line $\mathbf{l}_h$ and the normal of the edgel $\mathbf{t}$ point into the same direction

(see figure 2.29, right). Using again the matrices (cf. (2.43))

$$C_\infty = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{2.330}$$

and (cf. (2.46))

$$R_\perp = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.331}$$

the parallelity of the line and the edgel is defined in complete analogy to equation (2.309) as (see figure 2.29, right)

$$\mathsf{Parallel}(\mathbf{l}, e) \Leftrightarrow \mathbf{l}^T R_\perp C_\infty \mathbf{t} = \mathbf{t}^T C_\infty^T R_\perp^T \mathbf{l} \stackrel{!}{=} 0 \tag{2.332}$$

and hence incidence between a line and an edgel can be defined as

$$\mathsf{Incident}(\mathbf{l}, e) \Leftrightarrow \mathsf{Incident}(\mathbf{x}, \mathbf{l}) \wedge \mathsf{Parallel}(\mathbf{l}, e) \tag{2.333}$$

The previous definition straightforwardly extends toward 2d line segments in line representation

$$s : (\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}, \{\mathsf{C}_{ll}, \mathsf{C}_{mm}, \mathsf{C}_{nn}\}) \tag{2.334}$$

so that a line segment and an edgel are considered incident, if the following condition holds (see figure 2.29, right)

$$\mathsf{Incident}(s, e) \Leftrightarrow \mathsf{Incident}(\mathbf{x}, s) \wedge \mathsf{Parallel}(\mathbf{l}, e) \tag{2.335}$$

It is also possible to test, if a line pierces orthogonal through an edgel. Therefore one first defines in analogy to equation (2.298) orthogonality between a line and an edgels normal as

$$\mathsf{Orthogonal}(\mathbf{l}, e) \Leftrightarrow \mathbf{l}^T C_\infty \mathbf{t} = \mathbf{t}^T C_\infty^T \mathbf{l} \stackrel{!}{=} 0 \tag{2.336}$$

Now one can define an edgel and a line as being orthogonal, if (see figure 2.29, right)

$$\mathsf{Orthogonal}(\mathbf{l}, e) \Leftrightarrow \mathsf{Incident}(\mathbf{x}, \mathbf{l}) \wedge \mathsf{Orthogonal}(\mathbf{l}, e) \tag{2.337}$$

As before, this notion straightforwardly extends toward 2d line segments as (see figure 2.29, right)

$$\mathsf{Orthogonal}(s, e) \Leftrightarrow \mathsf{Incident}(\mathbf{x}, s) \wedge \mathsf{Orthogonal}(\mathbf{l}, e) \tag{2.338}$$

Two edgels are considered equal, if the points and the normals are identical, so that identity of the two uncertain oriented projective 2d edgels

$$e_1 : (\{\mathbf{x}_1, \mathbf{t}_1\}, \{\mathsf{C}_{x_1x_1}, \mathsf{C}_{t_1t_1}\}) \tag{2.339}$$

and

$$e_2 : (\{\mathbf{x}_2, \mathbf{t}_2\}, \{\mathsf{C}_{x_2x_2}, \mathsf{C}_{t_2t_2}\}) \tag{2.340}$$

up to orientation is defined as

$$\mathsf{Equal}(e_1, e_2) \Leftrightarrow \mathsf{Equal}(\mathbf{x}_1, \mathbf{x}_2) \wedge \mathsf{Equal}(\mathbf{t}_1, \mathbf{t}_2) \tag{2.341}$$

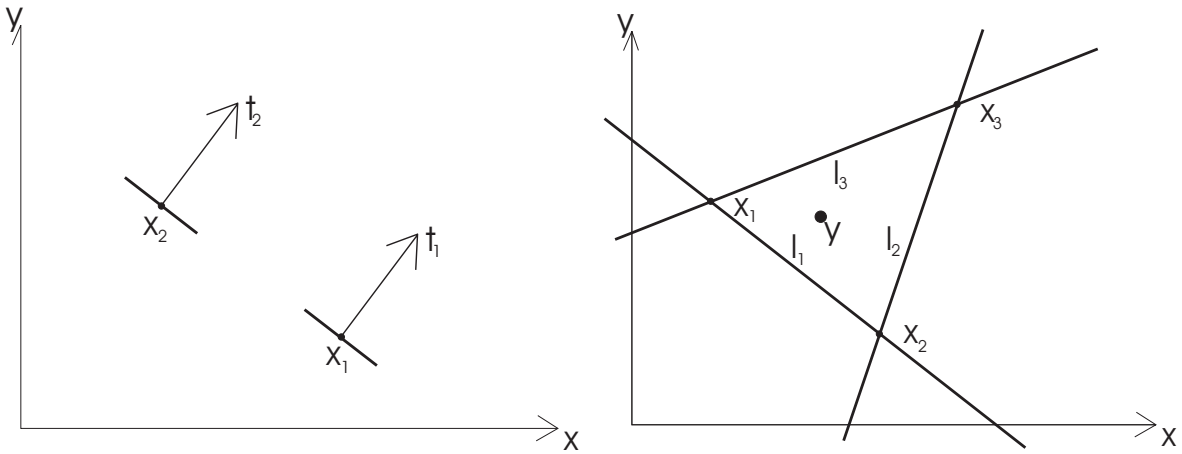

Figure 2.30: *Left:* The two edgels are considered aligned, because the two normal vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ point into the same direction and point $\mathbf{x}_1$ is on the line defined by the second edgel and vice versa. *Right:* The 2d point $\mathbf{y}$ is inside the 2d triangle, because it is left of all delimiting lines $\mathbf{l}_1$, $\mathbf{l}_2$ and $\mathbf{l}_3$.

where the equality up to orientation of two directions $\mathbf{t}_1$ and $\mathbf{t}_2$ is defined by requiring that the determinant $|\mathbf{t}_1 \quad \mathbf{t}_2|$ is zero, i.e.

$$\mathsf{Equal}(\mathbf{t}_1, \mathbf{t}_2) \Leftrightarrow \mathbf{t}_1^T \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{t}_2 = \mathbf{t}_2^T \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{t}_1 \stackrel{!}{=} 0 \tag{2.342}$$

Finally two edgels are aligned, if their normals point into the same direction and the position of one edgel lies on the line induced by the other edgel (see figure 2.30, left). Therefore alignment is defined as

$$\mathsf{Aligned}(e_1, e_2) \Leftrightarrow \mathsf{Equal}(\mathbf{t}_1, \mathbf{t}_2) \wedge \mathsf{Incident}(\mathbf{x}_1, \mathbf{l}_2) \tag{2.343}$$

with the line $\mathbf{l}_2$ constructed from the edgel e_2 according to equation (2.47).

2d polygons

Next are relations involving 2d polygons. In the following it will be assumed that all polygons are convex. If non-convex polygons are to be tested, it is required to break them up into convex sub-polygons, for instance using a Delaunay triangulation.

An uncertain oriented projective 2d point $(\mathbf{x}, \mathsf{C}_{xx})$ is inside a convex oriented projective 2d polygon in line representation

$$p : (\{\mathbf{l}_1, ..., \mathbf{l}_N\}, \{\mathsf{C}_{l_1 l_1}, ..., \mathsf{C}_{l_N l_N}\}) \tag{2.344}$$

if it lies left of all lines (see figure 2.30, right), i.e.

$$\mathsf{Inside}(\mathbf{x}, p) \Leftrightarrow \bigwedge_{i=1,...,N} \mathsf{Left}(\mathbf{x}, \mathbf{l}_i) \tag{2.345}$$

An uncertain oriented projective 2d line $(\mathbf{l}, \mathsf{C}_{ll})$ intersects the polygon, if it intersects any of the delimiting line segments. Hence, intersection with a convex uncertain oriented 2d polygon in point representation

$$p : (\{\mathbf{x}_1, ..., \mathbf{x}_N\}, \{\mathsf{C}_{x_1 x_1}, ..., \mathsf{C}_{x_N x_N}\}) \tag{2.346}$$

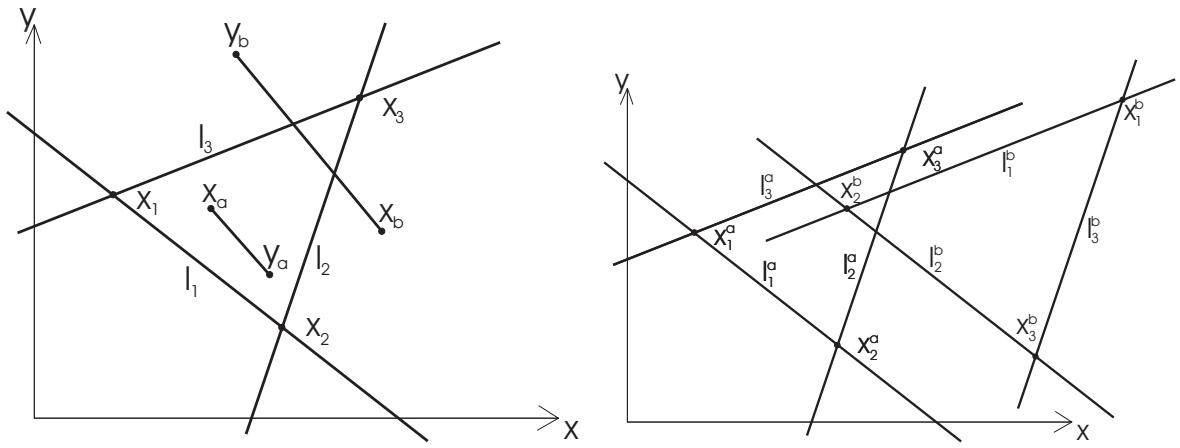

Figure 2.31: *Left:* The 2d line segment $\{\mathbf{x}_a, \mathbf{x}_b\}$ intersects the 2d triangle, because both its end-points are left of all the lines $\mathbf{l}_1$, $\mathbf{l}_2$ and $\mathbf{l}_3$. The 2d line segment $\{\mathbf{x}_b, \mathbf{x}_b\}$ intersects the 2d triangle, because it intersects the delimiting 2d line segment $\{\mathbf{x}_2, \mathbf{x}_3\}$ and also the delimiting 2d line segment $\{\mathbf{x}_3, \mathbf{x}_1\}$. *Right:* The two depicted 2d triangles intersect, because the point $\mathbf{x}_2^b$ is left of all delimiting lines $\mathbf{l}_1^a$, $\mathbf{l}_2^a$ and $\mathbf{l}_3^a$ and because the 2d line segments $\{\mathbf{x}_1^b, \mathbf{x}_2^b\}$ and $\{\mathbf{x}_2^b, \mathbf{x}_3^b\}$ intersect the 2d line segment $\{\mathbf{x}_2^a, \mathbf{x}_3^a\}$.

is defined as

$$\mathsf{Intersect}(\mathbf{l}, p) \Leftrightarrow \bigvee_{i=1,...,N} \mathsf{Opposite}(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{l}) \tag{2.347}$$

where the indices read modulo N.

An uncertain oriented projective 2d line segment in point representation as well as in line representation

$$\begin{aligned} s : (\{\mathbf{l}, \mathbf{m}, \mathbf{n}\}, \{\mathsf{C}_{ll}, \mathsf{C}_{mm}, \mathsf{C}_{nn}\}) \\ \leftrightarrow (\{\mathbf{x}, \mathbf{y}\}, \{\mathsf{C}_{xx}, \mathsf{C}_{yy}\}) \end{aligned} \tag{2.348}$$

intersects a convex uncertain oriented projective 2d polygon, if either both end-points are within the polygon or the line segment intersects any of the delimiting line segments of the polygon (see figure 2.31, left). Therefore intersection is defined in this case as

$$\begin{aligned} &\mathsf{Intersect}(s, p) \\ &\Leftrightarrow \left(\bigwedge_{i=1,...,N} \mathsf{Left}(\mathbf{x}, \mathbf{l}_i) \wedge \bigwedge_{i=1,...,N} \mathsf{Left}(\mathbf{y}, \mathbf{l}_i) \right) \vee \\ &\bigvee_{i=1,...,N} (\mathsf{Opposite}(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{l}) \wedge \mathsf{Opposite}(\mathbf{x}, \mathbf{y}, \mathbf{l}_i)) \end{aligned} \tag{2.349}$$

Two convex uncertain oriented projective 2d polygons in line as well as in point representation

$$\begin{aligned} p_1 \quad : \quad & (\{\mathbf{l}_1^{(1)}, ..., \mathbf{l}_N^{(1)}\}, \{\mathsf{C}_{l_1 l_1}^{(1)}, ..., \mathsf{C}_{l_N l_N}^{(1)}\}) \\ & \leftrightarrow (\{\mathbf{x}_1^{(1)}, ..., \mathbf{x}_N^{(1)}\}, \{\mathsf{C}_{x_1 x_1}^{(1)}, ..., \mathsf{C}_{x_N x_N}^{(1)}\}) \end{aligned} \tag{2.350}$$

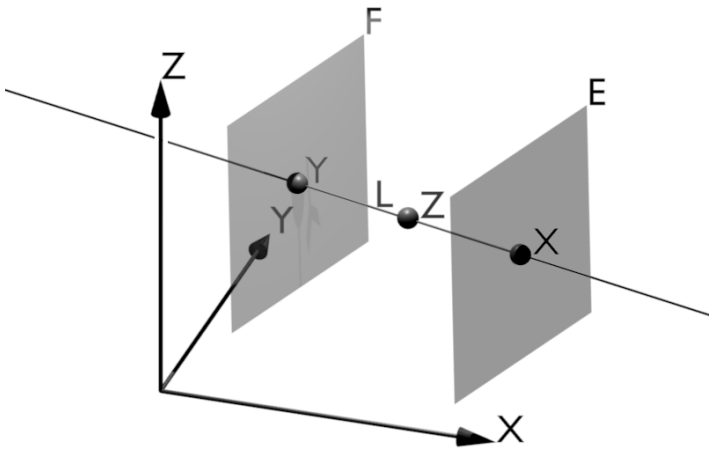

Figure 2.32: The 3d point **Z** is incident to the depicted 3d line segment, because it is incident to the line **L** and between the two delimiting planes **E** and **F**.

and

$$\begin{aligned} p_2 \quad : \quad & (\{\mathbf{l}_1^{(2)}, ..., \mathbf{l}_M^{(2)}\}, \{\mathsf{C}_{l_1 l_1}^{(2)}, ..., \mathsf{C}_{l_M l_M}^{(2)}\}) \\ & \leftrightarrow (\{\mathbf{x}_1^{(2)}, ..., \mathbf{x}_M^{(2)}\}, \{\mathsf{C}_{x_1 x_1}^{(2)}, ..., \mathsf{C}_{x_M x_M}^{(2)}\}) \end{aligned} \tag{2.351}$$

intersect, if either one polygon lies within the other or any of the delimiting line segments intersect (see figure 2.31, right). Therefore intersection is defined in this case as

$$\begin{aligned} & \mathsf{Intersect}(p_1, p_2) \\ & \Leftrightarrow \left(\bigwedge_{i=1,...,N} \bigwedge_{j=1,...,M} \mathsf{Left}(\mathbf{x}_i^{(1)}, \mathbf{l}_j^{(2)}) \right) \vee \left(\bigwedge_{i=1,...,M} \bigwedge_{j=1,...,N} \mathsf{Left}(\mathbf{x}_i^{(2)}, \mathbf{l}_j^{(1)}) \right) \vee \\ & \bigvee_{i=1,...,N} \bigvee_{j=1,...,M} (\mathsf{Opposite}(\mathbf{x}_i^{(1)}, \mathbf{x}_{i+1}^{(1)}, \mathbf{l}_j^{(2)}) \wedge \mathsf{Opposite}(\mathbf{x}_j^{(2)}, \mathbf{x}_{j+1}^{(2)}, \mathbf{l}_i^{(1)})) \end{aligned} \tag{2.352}$$

Observe that the number of corner points does not need to be equal.

The polygons are considered equal, if for some circulation of indices the endpoints are equal. Hence, it is defined as

$$\mathsf{Equal}(p_1, p_2) \Leftrightarrow \bigvee_{i=1,...,N} \bigwedge_{j=1,...,N} \mathsf{Equal}(\mathbf{x}_{j+i}^{(1)}, \mathbf{x}_j^{(2)}) \tag{2.353}$$

Observe that this test does not required convexity. Of course the number of corner points must be equal, though.

3d line segments

The relations for 2d line segments carry over to 3d line segments in an analogous manner and will be discussed in detail in the following.

Incidence: Given an uncertain oriented 3d line segment in line plane representation

$$\mathcal{S} : (\{\mathbf{L}, \mathbf{E}, \mathbf{F}\}, \{\mathsf{C}_{LL}, \mathsf{C}_{EE}, \mathsf{C}_{FF}\}) \tag{2.354}$$

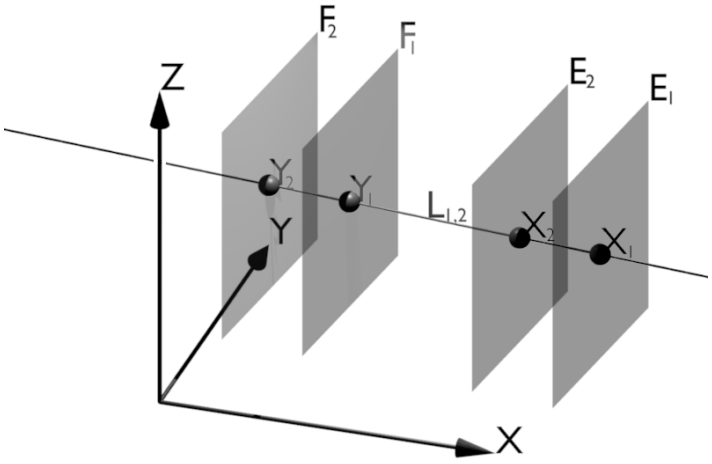

Figure 2.33: The two depicted 3d line segments overlap, because the two lines $\mathbf{L}_1$ and $\mathbf{L}_2$ are identical and the 3d point $\mathbf{X}_2$ is between the two 3d planes $\mathbf{E}_1$ and $\mathbf{F}_1$ and the 3d point $\mathbf{X}_1$ is between the two 3d planes $\mathbf{E}_2$ and $\mathbf{F}_2$.

an uncertain oriented 3d point $(\mathbf{X}, \mathsf{C}_{XX})$ lies on the line segment, if it is on the line $\mathbf{L}$ and between the planes $\mathbf{E}$ and $\mathbf{F}$ (see figure 2.32). Introducing the 3d version of the 'between' relation (cf. (2.318))

$$\mathsf{Between}(\mathbf{X}, \mathbf{E}, \mathbf{F}) \Leftrightarrow \mathsf{Below}(\mathbf{X}, \mathbf{E}) \wedge \mathsf{Below}(\mathbf{X}, \mathbf{F}) \tag{2.355}$$

incidence is defined analogously to the 2d case as

$$\mathsf{Incident}(\mathbf{X}, \mathcal{S}) \Leftrightarrow \mathsf{Incident}(\mathbf{X}, \mathbf{L}) \wedge \mathsf{Between}(\mathbf{X}, \mathbf{E}, \mathbf{F}) \tag{2.356}$$

For two uncertain oriented 3d line segments in line-plane representation as well as in point representation

$$\begin{aligned} \mathcal{S}_1 : (\{\mathbf{L}_1, \mathbf{E}_1, \mathbf{F}_1\}, \{\mathsf{C}_{L_1L_1}, \mathsf{C}_{E_1E_1}, \mathsf{C}_{F_1F_1}\}) \\ \leftrightarrow (\{\mathbf{X}_1, \mathbf{Y}_1\}, \{\mathsf{C}_{X_1X_1}, \mathsf{C}_{Y_1Y_1}\}) \end{aligned} \tag{2.357}$$

and

$$\begin{aligned} \mathcal{S}_2 : (\{\mathbf{L}_2, \mathbf{E}_2, \mathbf{F}_2\}, \{\mathsf{C}_{L_2L_2}, \mathsf{C}_{E_2E_2}, \mathsf{C}_{F_2F_2}\}) \\ \leftrightarrow (\{\mathbf{X}_2, \mathbf{Y}_2\}, \{\mathsf{C}_{X_2X_2}, \mathsf{C}_{Y_2Y_2}\}) \end{aligned} \tag{2.358}$$

overlap is defined analogously to the 2d case as (cf. (2.322) and see figure 2.33)

$$\begin{aligned} &\mathsf{Overlap}(\mathcal{S}_1, \mathcal{S}_2) \\ &\Leftrightarrow \mathsf{Equal}(\mathbf{L}_1, \mathbf{L}_2) \wedge (\mathsf{Between}(\mathbf{X}_1, \mathbf{E}_2, \mathbf{F}_2) \vee \mathsf{Between}(\mathbf{Y}_1, \mathbf{E}_2, \mathbf{F}_2) \vee \\ &\quad \mathsf{Between}(\mathbf{X}_2, \mathbf{E}_1, \mathbf{F}_1) \vee \mathsf{Between}(\mathbf{Y}_2, \mathbf{E}_1, \mathbf{F}_1)) \end{aligned} \tag{2.359}$$

Parallelity: Again, parallelity is defined for 3d line segments in contrast to infinite 3d lines by requiring the connecting lines not only to be parallel, but also the segments to be

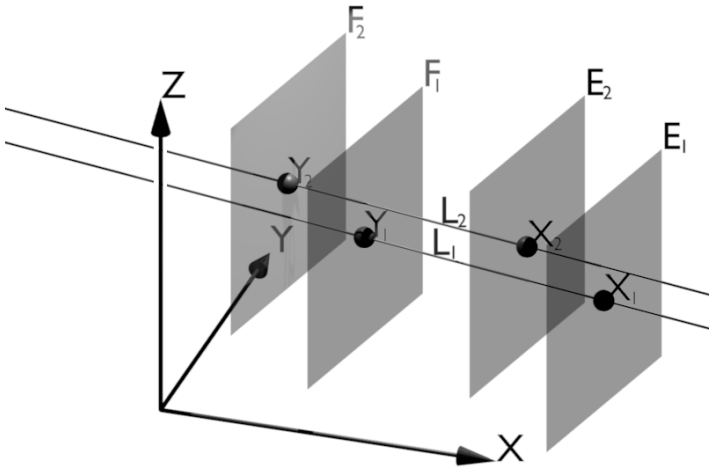

Figure 2.34: The two depicted 3d line segments are parallel, because the two lines $\mathbf{L}_1$ and $\mathbf{L}_2$ are parallel and the 3d point $\mathbf{X}_2$ is between the two 3d planes $\mathbf{E}_1$ and $\mathbf{F}_1$ and the 3d point $\mathbf{X}_1$ is between the two 3d planes $\mathbf{E}_2$ and $\mathbf{F}_2$.

aligned (cf. (2.323) and see figure 2.34). Therefore it is defined as

$$\begin{aligned} &\mathsf{Parallel}(\mathcal{S}_1, \mathcal{S}_2) \\ &\Leftrightarrow \mathsf{Parallel}(\mathbf{L}_1, \mathbf{L}_2) \wedge (\mathsf{Between}(\mathbf{X}_1, \mathbf{E}_2, \mathbf{F}_2) \vee \mathsf{Between}(\mathbf{Y}_1, \mathbf{E}_2, \mathbf{F}_2) \vee \\ &\qquad \mathsf{Between}(\mathbf{X}_2, \mathbf{E}_1, \mathbf{F}_1) \vee \mathsf{Between}(\mathbf{Y}_2, \mathbf{E}_1, \mathbf{F}_1)) \end{aligned} \tag{2.360}$$

Intersection: To define intersection of 3d line segments, the 'opposite' relation (cf. (2.324)) has to be extended for points with respect to a line. This is a little more involved, since the 3d line is not the dual entity of the 3d point. In order to solve this problem, the signs of the incidence relation between projective 3d points and projective 3d lines (cf. (2.290)) are analyzed. Those signs may be interpreted as indicators, whether the point lies left or right of the canonical entities of the line (cf. Definition 6 in [Heuel, 2004, p.67]), which are 3d planes and therefore dual entities of 3d points. Denoting the strongest canonical plane contained in the rows of the reduced $\overline{\mathbf{\Gamma}}$-matrix in equation (2.290) with

$$\overline{\boldsymbol{\gamma}}(\mathbf{L}) = \left(\left(\overline{\mathbf{\Gamma}}(\mathbf{L})^T \right)^{[1]} \right)^T \tag{2.361}$$

then two points $\mathbf{X}$ and $\mathbf{Y}$ lie on opposite sides of the line $\mathbf{L}$, if they lie on opposite sides of this plane $\overline{\boldsymbol{\gamma}}(\mathbf{L})$. First the 'opposite' relation for points $\mathbf{X}$ and $\mathbf{Y}$ with respect to a plane $\mathbf{A}$ is abbreviated as

$$\begin{aligned} \mathsf{Opposite}(\mathbf{X}, \mathbf{Y}, \mathbf{A}) \quad \Leftrightarrow \quad &(\mathsf{Above}(\mathbf{X}, \mathbf{A}) \wedge \mathsf{Below}(\mathbf{Y}, \mathbf{A})) \vee \\ &(\mathsf{Below}(\mathbf{X}, \mathbf{A}) \wedge \mathsf{Above}(\mathbf{Y}, \mathbf{A})) \end{aligned} \tag{2.362}$$

so that now the 'opposite' relation for points with respect to a line is easily formulated as

$$\mathsf{Opposite}(\mathbf{X}, \mathbf{Y}, \mathbf{L}) \Leftrightarrow \mathsf{Opposite}(\mathbf{X}, \mathbf{Y}, \overline{\boldsymbol{\gamma}}(\mathbf{L})) \tag{2.363}$$

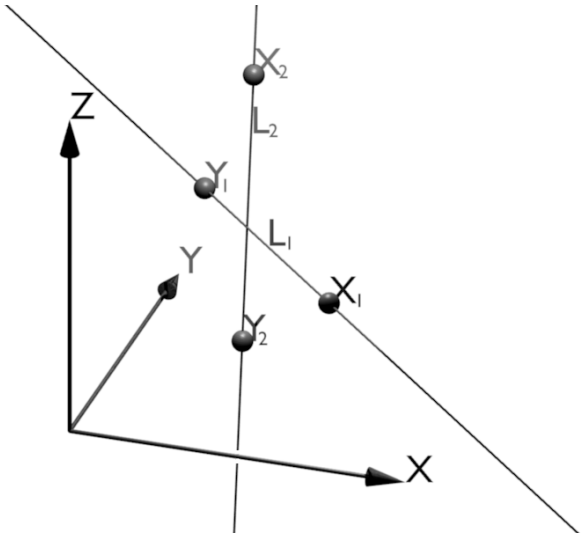

Figure 2.35: The line segment $\{\mathbf{X}_1, \mathbf{Y}_1\}$ intersects the line $\mathbf{L}_2$, because the two 3d lines $\mathbf{L}_1$ and $\mathbf{L}2$ intersect and the two 3d points $\mathbf{X}_1$ and $\mathbf{Y}_1$ are on opposite sides of the 3d line $\mathbf{L}_2$. The two depicted 3d line segments intersect, because the two lines $\mathbf{L}_1$ and $\mathbf{L}_2$ intersect and the two 3d points $\mathbf{X}_1$ and $\mathbf{Y}_1$ are on opposite sides of $\mathbf{L}_2$ and the two 3d points $\mathbf{X}_2$ and $\mathbf{Y}_2$ are on opposite sides of $\mathbf{L}_1$. Observe that the opposite relation between points and lines is only well defined, because the two lines intersect.

Note that (although not explicitly visible in the chosen notation) the covariance matrices of the canonical plane $\mathsf{C}_{\overline{\gamma}(L)\overline{\gamma}(L)}$ is required to define this relation. Fortunately, it is easily obtainable by selecting the respective rows and columns of the covariance matrix of the line.

Also observe that the given 'opposite' relation is only interpretable, if the line and the two points are co-planar. Therefore the intersection of an uncertain oriented projective 3d line segment with an uncertain oriented projective 3d line $\mathbf{L}$ is defined as (see figure 2.35)

$$\mathsf{Intersect}(\mathcal{S}_1, \mathbf{L}) \Leftrightarrow \mathsf{Incident}(\mathbf{L}_1, \mathbf{L}) \wedge \mathsf{Opposite}(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{L}) \tag{2.364}$$

The definition is extendable for two uncertain oriented projective 3d line segments by requiring that both segments intersect the other connecting line, yielding (see figure 2.35)

$$\begin{aligned} &\mathsf{Intersect}(\mathcal{S}_1, \mathcal{S}_2) \\ &\quad \Leftrightarrow \mathsf{Incident}(\mathbf{L}_1, \mathbf{L}_2) \wedge \mathsf{Opposite}(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{L}_2) \wedge \mathsf{Opposite}(\mathbf{X}_2, \mathbf{Y}_2, \mathbf{L}_1) \end{aligned} \tag{2.365}$$

Orthogonality: As in the 2d case the intersection relation can be extended toward orthogonal intersection, which yields the definition for the 3d case

$$\begin{aligned} &\mathsf{Orthogonal}(\mathcal{S}_1, \mathcal{S}_2) \\ &\quad \Leftrightarrow \mathsf{Orthogonal}(\mathbf{L}_1, \mathbf{L}_2) \wedge \mathsf{Incident}(\mathbf{L}_1, \mathbf{L}_2) \\ &\qquad \wedge \mathsf{Opposite}(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{L}_2) \wedge \mathsf{Opposite}(\mathbf{X}_2, \mathbf{Y}_2, \mathbf{L}_1) \end{aligned} \tag{2.366}$$

Identity: Finally identity of two directed uncertain oriented projective 3d line segments is definable through the identity of their end-points as

$$\mathsf{Equal}(\mathcal{S}_1, \mathcal{S}_2) \Leftrightarrow \mathsf{Equal}(\mathbf{X}_1, \mathbf{X}_2) \wedge \mathsf{Equal}(\mathbf{Y}_1, \mathbf{Y}_2) \tag{2.367}$$

Note that again like in the 2d case a 3d line segment passing the plane at infinity can be considered equal to a 3d line segment, if their end-points are equal up to orientation.

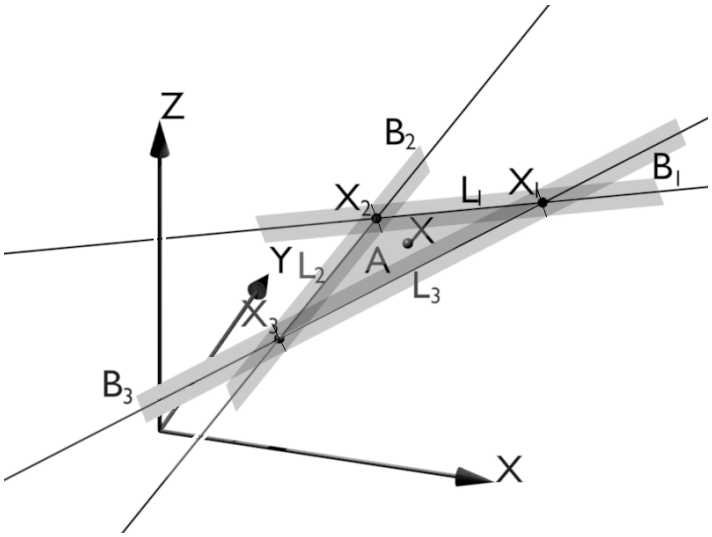

Figure 2.36: The 3d point $\mathbf{X}$ is inside the depicted 3d triangle, because it is on the plane $\mathbf{A}$ and above all delimiting planes $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$.

3d polygons

The relations for the 2d polygons carry over to the 3d case. Again convexity is assumed for all relations, unless stated otherwise.

An uncertain oriented 3d point $(\mathbf{X}, \mathsf{C}_{XX})$ lies within a convex uncertain oriented 3d polygon in plane representation

$$\mathcal{P} : (\{\mathbf{A}, \mathbf{B}_1, ..., \mathbf{B}_N\}, \{\mathsf{C}_{AA}, \mathsf{C}_{B_1B_1}, ..., \mathsf{C}_{B_NB_N}\}) \tag{2.368}$$

if it lies on the polygonal plane and above of all delimiting planes (see figure 2.36). Therefore this relation is defined as

$$\mathsf{Inside}(\mathbf{X}, \mathcal{P}) \Leftrightarrow \mathsf{Incident}(\mathbf{X}, \mathbf{A}) \wedge \bigwedge_{i=1,...,N} \mathsf{Above}(\mathbf{X}, \mathbf{B}_i) \tag{2.369}$$

An uncertain projective 3d line $(\mathbf{L}, \mathsf{C}_{LL})$ pierces through the uncertain convex polygon in line representation

$$\mathcal{P} : (\{\mathbf{L}_1, ..., \mathbf{L}_N\}, \{\mathsf{C}_{L_1L_1}, ..., \mathsf{C}_{L_N,L_N}\}) \tag{2.370}$$

if it goes either left or right of all lines, depending on whether it pierces the polygon from below of from above (see figure 2.37). Intersection is therefore defined in this case as (cf. [Förstner and Wrobel, 2004, p.140])

$$\mathsf{Intersect}(\mathbf{L}, \mathcal{P}) \Leftrightarrow \left(\bigwedge_{i=1,...,N} \mathsf{Left}(\mathbf{L}, \mathbf{L}_i) \right) \vee \left(\bigwedge_{i=1,...,N} \mathsf{Right}(\mathbf{L}, \mathbf{L}_i) \right) \tag{2.371}$$

This definition is easily augmented by requiring that the line pierces the polygon orthogonally, i.e.

$$\begin{aligned} &\mathsf{Orthogonal}(\mathbf{L}, \mathcal{P}) \\ &\Leftrightarrow \mathsf{Orthogonal}(\mathbf{L}, \mathbf{A}) \wedge \left(\left(\bigwedge_{i=1,...,N} \mathsf{Left}(\mathbf{L}, \mathbf{L}_i) \right) \vee \left(\bigwedge_{i=1,...,N} \mathsf{Right}(\mathbf{L}, \mathbf{L}_i) \right) \right) \end{aligned} \tag{2.372}$$

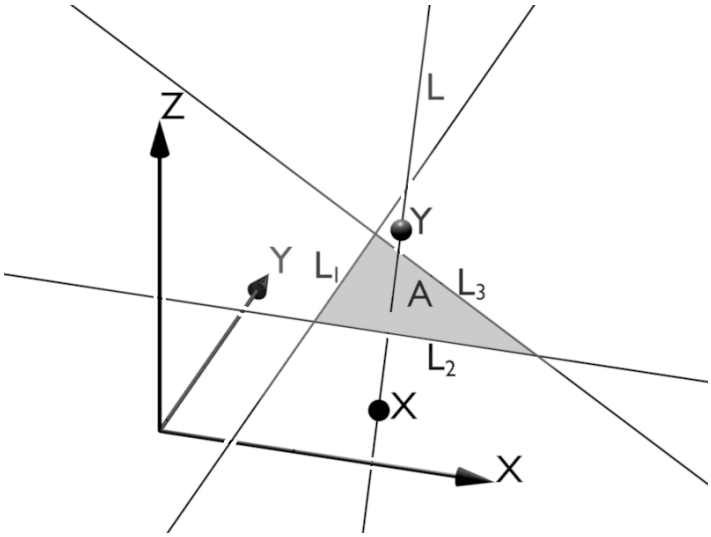

Figure 2.37: The line $\mathbf{L}$ intersects the depicted 3d triangle, because it is on the same side of all delimiting lines $\mathbf{L}_1$, $\mathbf{L}_2$ and $\mathbf{L}_3$. In addition to this, the 3d line segment $\{\mathbf{X}, \mathbf{Y}\}$ intersects the 3d polygon, because also the two 3d points $\mathbf{X}$ and $\mathbf{Y}$ are on opposite sides of $\mathbf{A}$.

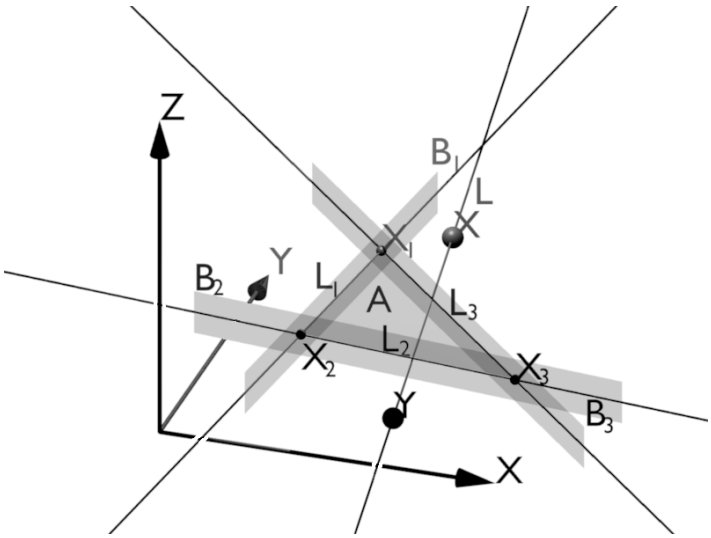

Figure 2.38: The 3d line $\mathbf{L}$ is incident to the depicted 3d polygon, because it is incident to the 3d plane $\mathbf{A}$ and the two 3d points $\mathbf{X}_1$ and $\mathbf{X}_2$ are on the opposite side of $\mathbf{L}$ than the 3d point $\mathbf{X}_3$. Note that the notion of two 3d points being opposite to each other with respect to a 3d line is only defined, because the points and the line are co-planar. In addition to this the 3d line segment $\{\mathbf{X}, \mathbf{Y}\}$ is incident to the 3d triangle, because also the points $\mathbf{X}$ and $\mathbf{Y}$ are opposite to each other with respect to the 3d lines $\mathbf{L}_2$ and $\mathbf{L}_3$.

The uncertain projective 3d line is considered incident to the convex 3d polygon, if it is incident to the polygonal plane and crosses any of the delimiting line segments (see figure 2.38). Hence incidence is defined as

$$\mathsf{Incident}(\mathbf{L}, \mathcal{P}) \Leftrightarrow \mathsf{Incident}(\mathbf{L}, \mathbf{A}) \wedge \bigvee_{i=1,\ldots,N} \mathsf{Opposite}(\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{L}) \tag{2.373}$$

where again all indices read modulo N. There is no easy generalization toward parallelity, because the 'opposite' relation requires the lines to intersect to be meaningful.

An uncertain oriented projective 3d line segment in point as wall as in line-plane representation

$$\begin{aligned} \mathcal{S} : (\{\mathbf{L}, \mathbf{E}, \mathbf{F}\}, \{\mathsf{C}_{LL}, \mathsf{C}_{EE}, \mathsf{C}_{FF}\}) \\ \leftrightarrow (\{\mathbf{X}, \mathbf{Y}\}, \{\mathsf{C}_{XX}, \mathsf{C}_{YY}\}) \end{aligned} \tag{2.374}$$

intersects a convex uncertain oriented projective 3d polygon, if the connecting line pierces the polygon and both end-points are on opposite sides of the polygonal plane (see figure 2.37). Hence, intersection is defined as

$$\begin{aligned} &\mathsf{Intersect}(\mathcal{S}, \mathcal{P}) \\ &\quad \Leftrightarrow \mathsf{Opposite}(\mathbf{X}, \mathbf{Y}, \mathbf{A}) \wedge \\ &\qquad \left(\left(\bigwedge_{i=1,\ldots,N} \mathsf{Left}(\mathbf{L}, \mathbf{L}_i) \right) \vee \left(\bigwedge_{i=1,\ldots,N} \mathsf{Right}(\mathbf{L}, \mathbf{L}_i) \right) \right) \end{aligned} \tag{2.375}$$

Again this definition is easily augmented by requiring the line segment to pierce the polygon orthogonally. Therefore this is defined as

$$\begin{aligned} &\mathsf{Orthogonal}(\mathcal{S}, \mathcal{P}) \\ &\quad \Leftrightarrow \mathsf{Opposite}(\mathbf{X}, \mathbf{Y}, \mathbf{A}) \wedge \mathsf{Orthogonal}(\mathbf{L}, \mathbf{A}) \wedge \\ &\qquad \left(\left(\bigwedge_{i=1,\ldots,N} \mathsf{Left}(\mathbf{L}, \mathbf{L}_i) \right) \vee \left(\bigwedge_{i=1,\ldots,N} \mathsf{Right}(\mathbf{L}, \mathbf{L}_i) \right) \right) \end{aligned} \tag{2.376}$$

The line segment is considered incident, if it is incident to the polygonal plane and either both end-points lie within the polygon or the segment intersects any of the delimiting segments (see figure 2.38).

$$\begin{aligned} &\mathsf{Incident}(\mathcal{S}, \mathcal{P}) \\ &\quad \Leftrightarrow \mathsf{Incident}(\mathbf{L}, \mathbf{A}) \wedge \Big(\left(\bigwedge_{i=1,\ldots,N} (\mathsf{Above}(\mathbf{X}, \mathbf{B}_i) \wedge \mathsf{Above}(\mathbf{Y}, \mathbf{B}_i)) \right) \vee \\ &\qquad \bigvee_{i=1,\ldots,N} (\mathsf{Opposite}(\mathbf{X}, \mathbf{Y}, \mathbf{B}_i) \wedge \mathsf{Opposite}(\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{L})) \Big) \end{aligned} \tag{2.377}$$

As before there is no easy generalization toward parallelity.

A convex uncertain projective 3d polygon intersects an uncertain projective plane $(\mathbf{D}, \mathsf{C}_{DD})$, if there exists some delimiting line segment, whose end-points are on both sides of the plane (see figure 2.39), i.e.

$$\mathsf{Intersect}(\mathbf{D}, \mathcal{P}) \Leftrightarrow \bigvee_{i=1,\ldots,N} \mathsf{Opposite}(\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{D}) \tag{2.378}$$

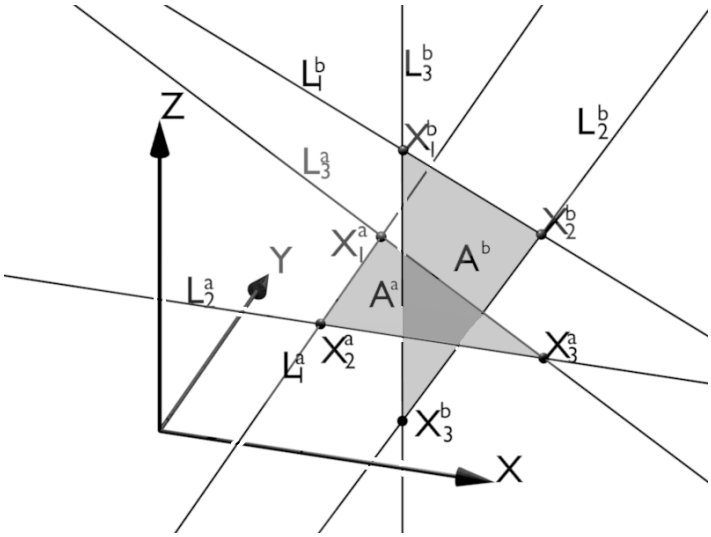

Figure 2.39: The 3d polygon $\{\mathbf{X}_1^a, \mathbf{X}_2^a, \mathbf{X}_3^a\}$ intersects the plane $\mathbf{A}^b$, because the 3d points $\mathbf{X}_1^a$ and $\mathbf{X}_3^a$ are on opposite sides of the 3d plane $\mathbf{A}^b$. The two depicted 3d triangles intersect, because the 3d line segment $\{qX_3^b, \mathbf{X}_1^b\}$ intersects the 3d polygon $\{\mathbf{X}_1^a, \mathbf{X}_2^a, \mathbf{X}_3^a\}$ and because the 3d line segment $\{\mathbf{X}_3^a, \mathbf{X}_1^a\}$ intersects the 3d polygon $\{\mathbf{X}_1^b, \mathbf{X}_2^b, \mathbf{X}_3^b\}$

.

This might also be augmented by orthogonality yielding

$$\begin{aligned} &\mathsf{Orthogonal}(\mathbf{D}, \mathcal{P}) \\ &\quad \Leftrightarrow \mathsf{Orthogonal}(\mathbf{A}, \mathbf{D}) \wedge \left(\bigvee_{i=1,...,N} \mathsf{Opposite}(\mathbf{X}_i, \mathbf{X}_{i+1}, \mathbf{D}) \right) \end{aligned} \tag{2.379}$$

Next, two convex uncertain projective 3d polygons in point, line and plane representation

$$\begin{aligned} \mathcal{P}_1 \quad : \quad & (\{\mathbf{X}_1^{(1)}, ..., \mathbf{X}_N^{(1)}\}, \{\mathsf{C}_{X_1X_1}^{(1)}, ..., \mathsf{C}_{X_NX_N}^{(1)}\}) \\ & \leftrightarrow (\{\mathbf{L}_1^{(1)}, ..., \mathbf{L}_N^{(1)}\}, \{\mathsf{C}_{L_1L_1}^{(1)}, ..., \mathsf{C}_{L_NL_N}^{(1)}\}) \\ & \leftrightarrow (\{\mathbf{A}^{(1)}, \mathbf{B}_1^{(1)}, ..., \mathbf{B}_N^{(1)}\}, \{\mathbf{C}_{AA}^{(1)}, \mathsf{C}_{B_1B_1}^{(1)}, ..., \mathsf{C}_{B_NB_N}^{(1)}\}) \end{aligned} \tag{2.380}$$

and

$$\begin{aligned} \mathcal{P}_2 \quad : \quad & (\{\mathbf{X}_1^{(2)}, ..., \mathbf{X}_M^{(2)}\}, \{\mathsf{C}_{X_1X_1}^{(2)}, ..., \mathsf{C}_{X_MX_M}^{(2)}\}) \\ & \leftrightarrow (\{\mathbf{L}_1^{(2)}, ..., \mathbf{L}_M^{(2)}\}, \{\mathsf{C}_{L_1L_1}^{(2)}, ..., \mathsf{C}_{L_ML_M}^{(2)}\}) \\ & \leftrightarrow (\{\mathbf{A}^{(2)}, \mathbf{B}_1^{(2)}, ..., \mathbf{B}_M^{(2)}\}, \{\mathbf{C}_{AA}^{(2)}, \mathsf{C}_{B_1B_1}^{(2)}, ..., \mathsf{C}_{B_MB_M}^{(2)}\}) \end{aligned} \tag{2.381}$$

intersect, if any delimiting line segment of one polygon pierces the other polygon (see figure 2.39). Hence, intersection is defined in this case as

$$\begin{aligned} &\mathsf{Intersect}(\mathcal{P}_1, \mathcal{P}_2) \\ &\quad \Leftrightarrow \bigvee_{i=1,...,N} \left(\left(\bigwedge_{j=1,...,M} \mathsf{Left}(\mathbf{L}_i^{(1)}, \mathbf{L}_j^{(2)}) \right) \vee \left(\bigwedge_{j=1,...,M} \mathsf{Right}(\mathbf{L}_i^{(1)}, \mathbf{L}_j^{(2)}) \right) \right) \end{aligned} \tag{2.382}$$

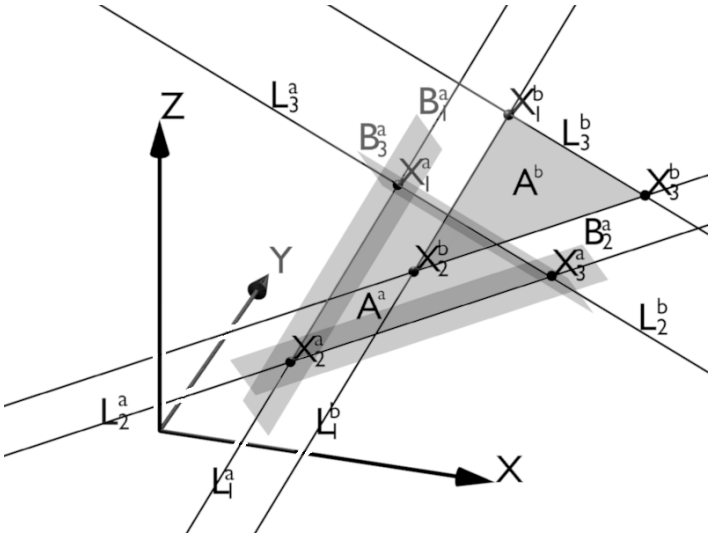

Figure 2.40: The two depicted 3d triangles overlap, because the two 3d planes $\mathbf{A}^a$ and $\mathbf{A}^b$ are identical and the 3d point $\mathbf{X}_2^b$ is above all the delimiting planes $\mathbf{B}_1^a$, $\mathbf{B}_2^a$ and $\mathbf{B}_3^a$.

$$\begin{aligned}
&\wedge \mathsf{Opposite}(\mathbf{X}_i^{(1)}, \mathbf{X}_{i+1}^{(1)}, \mathbf{A}^{(2)})) \\
\vee \bigvee_{i=1,\ldots,M} (&\left(\left(\bigwedge_{j=1,\ldots,N} \mathsf{Left}(\mathbf{L}_i^{(2)}, \mathbf{L}_j^{(1)}) \right) \vee \left(\bigwedge_{j=1,\ldots,N} \mathsf{Right}(\mathbf{L}_i^{(2)}, \mathbf{L}_j^{(1)}) \right) \right) \\
&\wedge \mathsf{Opposite}(\mathbf{X}_i^{(2)}, \mathbf{X}_{i+1}^{(2)}, \mathbf{A}^{(1)}))
\end{aligned}$$

Observe that the number of corner points does not need to be identical for this relation. Again, this definition can be extended toward orthogonal intersection yielding

$$\begin{aligned}
&\mathsf{Orthogonal}(\mathcal{P}_1, \mathcal{P}_2) \qquad (2.383) \\
&\quad \Leftrightarrow \mathsf{Orthogonal}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}) \wedge \\
&\quad (\bigvee_{i=1,\ldots,N} ((\left(\bigwedge_{j=1,\ldots,M} \mathsf{Left}(\mathbf{L}_i^{(1)}, \mathbf{L}_j^{(2)}) \right) \vee \left(\bigwedge_{j=1,\ldots,M} \mathsf{Right}(\mathbf{L}_i^{(1)}, \mathbf{L}_j^{(2)}) \right)) \\
&\quad \wedge \mathsf{Opposite}(\mathbf{X}_i^{(1)}, \mathbf{X}_{i+1}^{(1)}, \mathbf{A}^{(2)})) \\
&\quad \vee \bigvee_{i=1,\ldots,M} ((\left(\bigwedge_{j=1,\ldots,N} \mathsf{Left}(\mathbf{L}_i^{(2)}, \mathbf{L}_j^{(1)}) \right) \vee \left(\bigwedge_{j=1,\ldots,N} \mathsf{Right}(\mathbf{L}_i^{(2)}, \mathbf{L}_j^{(1)}) \right)) \\
&\quad \wedge \mathsf{Opposite}(\mathbf{X}_i^{(2)}, \mathbf{X}_{i+1}^{(2)}, \mathbf{A}^{(1)})))
\end{aligned}$$

The two convex 3d polygons are considered to overlap, if the polygonal planes are equal and either all points of one polygon lie within the other polygon or any of the delimiting line segments intersect (see figure 2.40). Therefore overlap is defined as

$$\begin{aligned}
&\mathsf{Overlap}(\mathcal{P}_1, \mathcal{P}_2) \qquad (2.384) \\
&\quad \Leftrightarrow \mathsf{Equal}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}) \wedge \\
&\quad (\left(\bigwedge_{i=1,\ldots,N} \bigwedge_{j=1,\ldots,M} \mathsf{Above}(\mathbf{X}_i^{(1)}, \mathbf{B}_j^{(2)}) \right) \vee
\end{aligned}$$

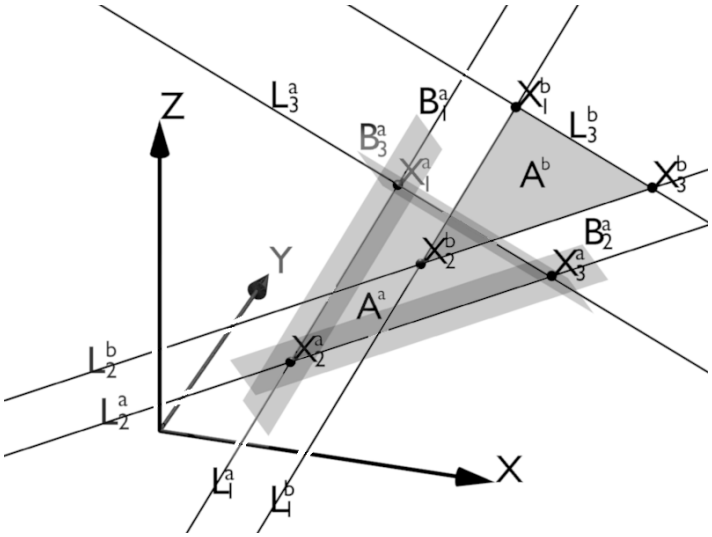

Figure 2.41: The two depicted 3d triangles are parallel, because the two 3d planes $\mathbf{A}^a$ and $\mathbf{A}^b$ are parallel and the 3d point $\mathbf{X}_2^b$ is above all the delimiting planes $\mathbf{B}_1^a$, $\mathbf{B}_2^a$ and $\mathbf{B}_3^a$.

$$\left(\bigwedge_{i=1,\ldots,M}\bigwedge_{j=1,\ldots,N}\mathsf{Above}(\mathbf{X}_i^{(2)},\mathbf{B}_j^{(1)})\right)\vee \bigvee_{i=1,\ldots,N}\bigvee_{j=1,\ldots,M}(\mathsf{Opposite}(\mathbf{X}_i^{(1)},\mathbf{X}_{i+1}^{(1)},\mathbf{B}_j^{(2)})\wedge \mathsf{Opposite}(\mathbf{X}_j^{(2)},\mathbf{X}_{j+1}^{(2)},\mathbf{B}_i^{(1)})))$$

Here the relation may be relaxed to paralellity of two convex 3d polygons (see figure 2.41) being

$$\begin{aligned}&\mathsf{Parallel}(\mathcal{P}_1,\mathcal{P}_2)\\ &\Leftrightarrow \mathsf{Parallel}(\mathbf{A}^{(1)},\mathbf{A}^{(2)})\wedge\\ &(\left(\bigwedge_{i=1,\ldots,N}\bigwedge_{j=1,\ldots,M}\mathsf{Above}(\mathbf{X}_i^{(1)},\mathbf{B}_j^{(2)})\right)\vee\\ &\left(\bigwedge_{i=1,\ldots,M}\bigwedge_{j=1,\ldots,N}\mathsf{Above}(\mathbf{X}_i^{(2)},\mathbf{B}_j^{(1)})\right)\vee\\ &\bigvee_{i=1,\ldots,N}\bigvee_{j=1,\ldots,M}(\mathsf{Opposite}(\mathbf{X}_i^{(1)},\mathbf{X}_{i+1}^{(1)},\mathbf{B}_j^{(2)})\wedge\\ &\qquad\mathsf{Opposite}(\mathbf{X}_j^{(2)},\mathbf{X}_{j+1}^{(2)},\mathbf{B}_i^{(1)})))\end{aligned}\tag{2.385}$$

As in the 2d case two polygons are considered equal, if for some circulation of indices the endpoints are equal. Hence, it is defined as

$$\mathsf{Equal}(\mathcal{P}_1,\mathcal{P}_2)\Leftrightarrow\bigvee_{i=1,\ldots,N}\bigwedge_{j=1,\ldots,N}\mathsf{Equal}(\mathbf{X}_{j+i}^{(1)},\mathbf{X}_j^{(2)})\tag{2.386}$$

Of course the number of end-points must be equal. Again this test does not required convexity.

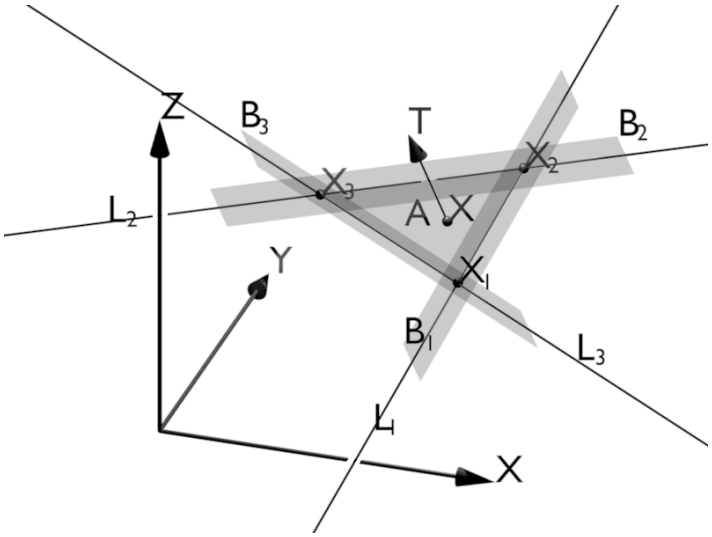

Figure 2.42: The 3d facet $\{\mathbf{X}, \mathbf{T}\}$ is incident to the plane $\mathbf{A}$, because the 3d point $\mathbf{X}$ is incident to $\mathbf{A}$ and the normal of $\mathbf{A}$ and $\mathbf{T}$ point into the same direction. In addition to this, the facet is incident to the depicted 3d triangle, because also the 3d point $\mathbf{X}$ is above all delimiting planes $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$.

3d facets

The relations for the 2d edgels carry over to the 3d case. An uncertain oriented projective 3d facet

$$\mathcal{F} : (\{\mathbf{X}, \mathbf{T}\}, \{\mathsf{C}_{XX}, \mathsf{C}_{TT}\}) \tag{2.387}$$

is considered incident to an uncertain oriented projective plane $(\mathbf{A}, \mathsf{C}_{AA})$, if the normal direction of the facet $\mathbf{T}$ and the normal direction of the plane $\boldsymbol{A}_h$ point in the same direction and the point $\mathbf{X}$ lies on the plane $\mathbf{A}$ (see figure 2.42). Being more specific and defining parallelity between a facet and a plane as (cf. (2.315))

$$\mathsf{Parallel}(\mathcal{F}, \mathbf{A}) \Leftrightarrow \begin{pmatrix} \mathsf{S}(\mathbf{T}) & \mathbf{0} \end{pmatrix}^{[2]} \mathbf{A} = -\left(\mathsf{S}(\boldsymbol{A}_h)\right)^{[2]} \mathbf{T} \stackrel{!}{=} \mathbf{0} \tag{2.388}$$

the incidence of a plane and a facet is defined as

$$\mathsf{Incident}(\mathcal{F}, \mathbf{A}) \Leftrightarrow \mathsf{Incident}(\mathbf{X}, \mathbf{A}) \wedge \mathsf{Parallel}(\mathcal{F}, \mathbf{A}) \tag{2.389}$$

The previous definition straightforwardly extends toward convex 3d polygons in plane representation

$$\mathcal{P} : (\{\mathbf{A}, \mathbf{B}_1, ..., \mathbf{B}_N\}, \{\mathsf{C}_{AA}, \mathsf{C}_{B_1B_1}, ..., \mathsf{C}_{B_NB_N}\}) \tag{2.390}$$

so that a convex 3d polygon and a facet are considered incident, if the following condition holds (see figure 2.42)

$$\mathsf{Incident}(\mathcal{F}, \mathcal{P}) \Leftrightarrow \mathsf{Inside}(\mathbf{X}, \mathcal{P}) \wedge \mathsf{Parallel}(\mathcal{F}, \mathbf{A}) \tag{2.391}$$

For a 3d line $\mathbf{L}$ and a facet orthogonality can be defined analogously. Therefore first a line and a facet are considered orthogonally aligned, if the direction of the line $\mathbf{L}_h$ and the normal $\mathbf{T}$ of the facet point in the same direction (see figure 2.43), i.e.

$$\begin{aligned} &\mathsf{OrthogonalAligned}(\mathcal{F}, \mathbf{L}) \\ &\Leftrightarrow \begin{pmatrix} \mathsf{S}(\mathbf{T}) & \mathit{0}_{3\times 3} \end{pmatrix}^{[2]} \mathbf{L} = -\left(\mathsf{S}(\boldsymbol{L}_h)\right)^{[2]} \mathbf{T} \stackrel{!}{=} \mathbf{0} \end{aligned} \tag{2.392}$$

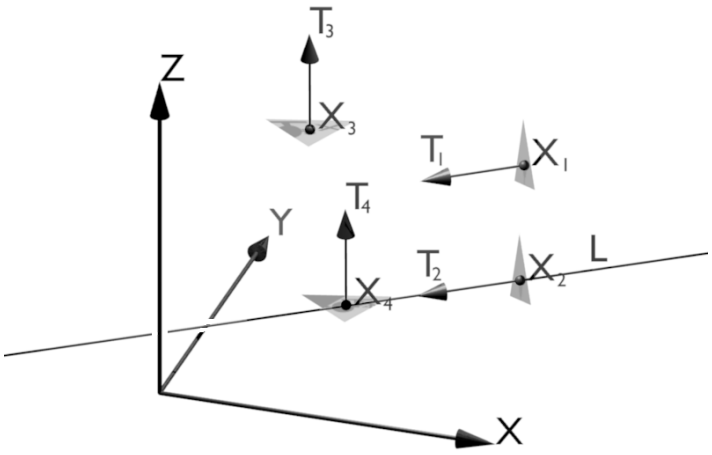

Figure 2.43: The 3d facet $\{\mathbf{X}_1, \mathbf{T}_1\}$ is orthogonal aligned to the 3d line $\mathbf{L}$, because its normal points into the same direction as the direction of the line. Furthermore the facet $\{\mathbf{X}_2, \mathbf{T}_2\}$ is orthogonal to the line, because in addition to being orthogonal aligned the 3d point $\mathbf{X}_2$ is incident to $\mathbf{L}$. The 3d facet $\{\mathbf{X}_3, \mathbf{T}_3\}$ is parallel to the 3d line $\mathbf{L}$, because its normal and the direction of the line are orthogonal. In addition to this, the 3d facet $\{\mathbf{X}_4, \mathbf{T}_4\}$ is incident to the 3d line $\mathbf{L}$, because it is parallel to the line and the point $\mathbf{X}_4$ is incident to $\mathbf{L}$.

Now orthogonality of a facet and a line is defined by requiring that in addition to being orthogonally aligned the point $\mathbf{X}$ lies on the line (see figure 2.43). The definition for the orthogonality of a line and a facet is therefore given by

$$\begin{aligned} &\mathsf{Orthogonal}(\mathcal{F}, \mathbf{L}) \\ &\quad \Leftrightarrow \mathsf{Incident}(\mathbf{X}, \mathbf{L}) \wedge \mathsf{OrthogonalAligned}(\mathcal{F}, \mathbf{L}) \end{aligned} \tag{2.393}$$

As before this notion straightforwardly extends toward 3d line segments, so that a 3d line segment in line-plane-representation

$$\mathcal{S} : (\{\mathbf{L}, \mathbf{E}, \mathbf{F}\}, \{\mathsf{C}_{LL}, \mathsf{C}_{EE}, \mathsf{C}_{FF}\}) \tag{2.394}$$

is orthogonal to a 3d facet, if (see figure 2.43)

$$\begin{aligned} &\mathsf{Orthogonal}(\mathcal{S}, \mathcal{F}) \\ &\quad \Leftrightarrow \mathsf{Incident}(\mathbf{X}, \mathcal{S}) \wedge \mathsf{OrthogonalAligned}(\mathcal{F}, \mathbf{L}) \end{aligned} \tag{2.395}$$

A 3d facet is considered to be incident to a line, if the point lies on the line and the normal of the facet is parallel to the line, i.e. perpendicular to its direction (see figure 2.43). Therefore one first defines parallelity between a 3d facet and a 3d line as (cf. (2.312))

$$\mathsf{Parallel}(\mathcal{F}, \mathbf{L}) \Leftrightarrow \mathbf{L}^T \begin{pmatrix} I_3 \\ 0_{3\times 3} \end{pmatrix} \mathbf{T} = \mathbf{T}^T \begin{pmatrix} I_3 & 0_{3\times 3} \end{pmatrix} \mathbf{L} \stackrel{!}{=} 0 \tag{2.396}$$

Now incidence between a 3d facet and a 3d line is defined by requiring that also the point lies on the line, yielding

$$\mathsf{Incident}(\mathcal{F}, \mathbf{L}) \Leftrightarrow \mathsf{Incident}(\mathbf{X}, \mathbf{L}) \wedge \mathsf{Parallel}(\mathcal{F}, \mathbf{L}) \tag{2.397}$$

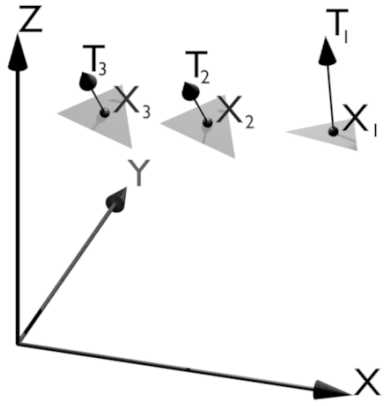

Figure 2.44: The 3d facets $\{\mathbf{X}_1, \mathbf{T}_1\}$ and $\{\mathbf{X}_2, \mathbf{T}_2\}$ are co-linear, because the 3d point $\mathbf{X}_1$ is on the plane defined by the second facet and vice versa. The 3d facets $\{\mathbf{X}_2, \mathbf{T}_2\}$ and $\{\mathbf{X}_3, \mathbf{T}_3\}$ are co-planar, because in addition to being co-linear, their normal vectors point into the same direction.

Again this definition can be easily extended toward 3d line segments, so that incidence is defined in this case as (see figure 2.43)

$$\mathsf{Incident}(\mathcal{F}, \mathcal{S}) \Leftrightarrow \mathsf{Incident}(\mathbf{X}, \mathcal{S}) \wedge \mathsf{Parallel}(\mathcal{F}, \mathbf{L}) \tag{2.398}$$

Equality between two 3d facets is defined by considering two 3d facets

$$\mathcal{F}_1 : (\{\mathbf{X}_1, \mathbf{T}_1\}, \{\mathsf{C}_{X_1X_1}, \mathsf{C}_{T_1T_1}\}) \tag{2.399}$$

and

$$\mathcal{F}_2 : (\{\mathbf{X}_2, \mathbf{T}_2\}, \{\mathsf{C}_{X_2X_2}, \mathsf{C}_{T_2T_2}\}) \tag{2.400}$$

as equal, if the points $\mathbf{X}_1$ and $\mathbf{X}_2$ as well as the normal directions T_1 and T_2 are equal up to orientation, i.e.

$$\mathsf{Equal}(\mathcal{F}_1, \mathcal{F}_2) \Leftrightarrow \mathsf{Equal}(\mathbf{X}_1, \mathbf{X}_2) \wedge \mathsf{Equal}(\mathbf{T}_1, \mathbf{T}_2) \tag{2.401}$$

where equality up to orientation between the two normal directions is defined analogously to the equality of two 2d points (cf. 2.292) as

$$\mathsf{Equal}(\mathbf{T}_1, \mathbf{T}_2) \Leftrightarrow (\mathcal{S}(\mathbf{T}_1))^{[2]}\, \mathbf{T}_2 = -\,(\mathcal{S}(\mathbf{T}_2))^{[2]}\, \mathbf{T}_1 \stackrel{!}{=} \mathbf{0} \tag{2.402}$$

Two facets are considered co-linear, if the first facets position $\mathbf{X}_1$ lies on the plane $\mathbf{A}_2$ defined by the second facet $\mathcal{F}_2$ according to equation (2.118) and vice versa (see figure 2.44). Co-linearity for two facets is therefore defined as

$$\mathsf{Colinear}(\mathcal{F}_1, \mathcal{F}_2) \Leftrightarrow \mathsf{Incident}(\mathbf{X}_1, \mathbf{A}_2) \wedge \mathsf{Incident}(\mathbf{X}_2, \mathbf{A}_1) \tag{2.403}$$

Finally two facets are considered to be co-planar, if their normals point into the same direction and the position of one facet lies on the plane induced by the other facet (see figure 2.44). Hence co-planarity is defined as

$$\mathsf{Coplanar}(\mathcal{F}_1, \mathcal{F}_2) \Leftrightarrow \mathsf{Equal}(\mathbf{T}_1, \mathbf{T}_2) \wedge \mathsf{Incident}(\mathbf{X}_1, \mathbf{A}_2) \tag{2.404}$$

Up to now a number of relations between the uncertain oriented projective entities have been presented. All those tests take two entities as input and return a yes/no-conjecture about their relation as output. In case that a large number of tests has to be performed, which is the case in most real-world applications, some efficiency considerations are in order. In the following section a data structure will be derived that exploits the specific structure of the bi-linear tests presented in this section and allows to perform those tests more efficiently on large sets of entities.

2.3 Data structures for efficient testing

In the previous section various relations between pairs of entities were presented. In real application huge amounts of such pairwise tests need to be performed. In order to facilitate efficient testing it is possible to exploit the special structure of the presented relations and derive a data structure that enables finding all elements out of a large database fulfilling a given relation with another given element.

In the following such a data-structure, which has been presented briefly in [Beder, 2004a], will be derived in detail. It is inspired by the idea of the R-tree (cf. [Guttman, 1984]) for the efficient retrieval of spatial objects, which uses bounding boxes as necessary conditions for speeding up searches, as well as balancing the tree similar to B-trees (cf. [Bayer and McCreight, 1972]) in order to guarantee a non-degenerate logarithmic depth.

The bounding boxes will be generalized toward general necessary conditions for the bi-linear test relations. It is also necessary to have a method for combining those necessary conditions into new ones to build up the tree structure. Both notions, which are rather technical in nature, will be presented in the following.

2.3.1 Necessary conditions

All presented relations consist of conjunctions and disjunctions of equations either in the form (2.270) or in the form (2.273). In the following necessary conditions for those relations will be derived. Repeating equation (2.270)

$$\boldsymbol{y}^T A(\boldsymbol{x})^T \left(A(\boldsymbol{x}) C_{yy} A(\boldsymbol{x})^T + B(\boldsymbol{y}) C_{xx} B(\boldsymbol{y})^T \right)^{-1} A(\boldsymbol{x}) \boldsymbol{y} \leq T_{\alpha,r} \tag{2.405}$$

and equation (2.273)

$$\frac{\boldsymbol{a}(\boldsymbol{x})\boldsymbol{y}}{\sqrt{\boldsymbol{a}(\boldsymbol{x}) C_{yy} \boldsymbol{a}(\boldsymbol{x})^T + \boldsymbol{b}(\boldsymbol{y}) C_{xx} \boldsymbol{b}(\boldsymbol{y})^T}} \leq \sqrt{T_{\alpha,1}} \tag{2.406}$$

a managable necessary condition for this to hold is obtained by first enlarging the denominator. Therefore the error ellipsoids are replaced by the smallest possible round error ellipsoids still containing the original ones. Being more specific and denoting with σ_x^2 the largest eigenvalue of

$$C_{xx} = U \begin{pmatrix} \sigma_x^2 & & \\ & \ddots & \\ & & * \end{pmatrix} U^T \tag{2.407}$$

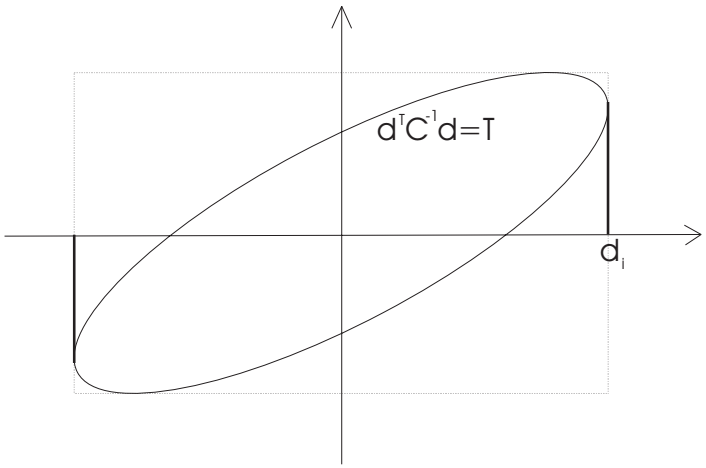

Figure 2.45: The bounding box of the ellipse defined by the solution to the quadratic equation $\boldsymbol{d}^T C^{-1}\boldsymbol{d} = T$ can be found by finding the point $\boldsymbol{d}$ on the ellipse, such that the i-th component d_i is maximum (see text).

and with σ_y^2 the largest eigenvalue of

$$C_{yy} = U \begin{pmatrix} \sigma_y^2 & & \\ & \ddots & \\ & & * \end{pmatrix} U^T \tag{2.408}$$

a necessary condition for (2.270) is given by

$$\boldsymbol{y}^T A(\boldsymbol{x})^T \left(\underbrace{\sigma_y^2 A(\boldsymbol{x})A(\boldsymbol{x})^T + \sigma_x^2 B(\boldsymbol{y})B(\boldsymbol{y})^T}_{C_{dd}} \right)^{-1} \underbrace{A(\boldsymbol{x})\boldsymbol{y}}_{\boldsymbol{d}} \tag{2.409}$$

$$= \quad \boldsymbol{d}^T C_{dd}^{-1} \boldsymbol{d} \leq T_{\alpha,r} \tag{2.410}$$

and a necessary condition for (2.273) is given by

$$\frac{\boldsymbol{a}(\boldsymbol{x})\boldsymbol{y}}{\sqrt{\sigma_y^2 \boldsymbol{a}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x})^T + \sigma_x^2 \boldsymbol{b}(\boldsymbol{y})\boldsymbol{b}(\boldsymbol{y})^T}} \leq \sqrt{T_{\alpha,1}} \tag{2.411}$$

The next step is to find a bounding box for the error ellipsoid represented by equation (2.410). Therefore the vector $\boldsymbol{d} = A(\boldsymbol{x})\boldsymbol{y}$ is projected onto its components

$$d_i = \boldsymbol{e}_i^T \boldsymbol{d} = \boldsymbol{e}_i^T A(\boldsymbol{x})\boldsymbol{y} = \boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{y} \tag{2.412}$$

where $\boldsymbol{e}_i$ denotes the unit basis vector containing zeros everywhere except for the i-th position, where it contains a one. In order to find this bounding box, the extremal value of d_i under the constraint that $\boldsymbol{d}$ lies on the ellipsoid

$$\boldsymbol{d}^T C_{dd}^{-1} \boldsymbol{d} = T \tag{2.413}$$

is determined using the technique of Lagrangian multipliers (see figure 2.45). Being more specific, the extrema of the function

$$f(\boldsymbol{d}, \lambda) = \boldsymbol{e}_i^T \boldsymbol{d} + \lambda(\boldsymbol{d}^T C_{dd}^{-1} \boldsymbol{d} - T) \tag{2.414}$$

are derived. Setting its partial derivative

$$\frac{\partial}{\partial \boldsymbol{d}} f(\boldsymbol{d}, \lambda) = \boldsymbol{e}_i^T + 2\lambda \boldsymbol{d}^T \mathsf{C}_{dd}^{-1} = \mathbf{0} \tag{2.415}$$

equal to zero and solving for $\boldsymbol{d}$ yields

$$\overline{\boldsymbol{d}} = \frac{\mathsf{C}\boldsymbol{e}_i}{2\lambda} \tag{2.416}$$

This solution is now inserted into the partial derivative

$$\begin{aligned} \frac{\partial}{\partial \lambda} f(\overline{\boldsymbol{d}}, \lambda) &= \overline{\boldsymbol{d}}^T \mathsf{C}_{dd}^{-1} \overline{\boldsymbol{d}} - T & (2.417)\\ &= \frac{\boldsymbol{e}_i^T \mathsf{C}\mathsf{C}^{-1}\mathsf{C}\boldsymbol{e}_i}{4\lambda^2} - T & (2.418)\\ &= \frac{\boldsymbol{e}_i^T \mathsf{C}\boldsymbol{e}_i}{4\lambda^2} - T = 0 & (2.419) \end{aligned}$$

Again, setting this equal to zero and solving for λ yields

$$\overline{\lambda} = \pm\sqrt{\frac{\boldsymbol{e}_i^T \mathsf{C}\boldsymbol{e}_i}{4T}} \tag{2.420}$$

Inserting this back into equation (2.416) yields the extremal value

$$\begin{aligned} \overline{\overline{\boldsymbol{d}}} &= \pm\frac{\mathsf{C}\boldsymbol{e}_i}{2\sqrt{\frac{\boldsymbol{e}_i^T \mathsf{C}\boldsymbol{e}_i}{4T}}} & (2.421)\\ &= \pm\sqrt{T}\frac{\mathsf{C}\boldsymbol{e}_i}{\sqrt{\boldsymbol{e}_i^T \mathsf{C}\boldsymbol{e}_i}} & (2.422) \end{aligned}$$

Hence, the extremal component is given by

$$\overline{\overline{d_i}} = \boldsymbol{e}_i^T \overline{\overline{\boldsymbol{d}}} = \pm\sqrt{T}\frac{\boldsymbol{e}_i^T \mathsf{C}\boldsymbol{e}_i}{\sqrt{\boldsymbol{e}_i^T \mathsf{C}\boldsymbol{e}_i}} \tag{2.423}$$

or equivalent

$$\frac{\overline{\overline{d_i}}^2}{C_{d_i d_i}} = \frac{\boldsymbol{e}_i^T \overline{\overline{d}}\, \overline{\overline{d}}^T \boldsymbol{e}_i}{\boldsymbol{e}_i^T C_{dd} \boldsymbol{e}_i} = T \tag{2.424}$$

Because the last equation holds for all T, it finally follows that

$$\boldsymbol{d}^T \mathsf{C}_{dd}^{-1} \boldsymbol{d} \leq T \Rightarrow \frac{d_i^2}{C_{d_i d_i}} \leq T \tag{2.425}$$

and that the bound is actually achieved for some $\boldsymbol{d}$ (cf. figure 2.14 in [Förstner and Wrobel, 2004, p.63]).

Hence, a necessary condition for equation (2.410) is that for all i the following equation holds

$$\frac{(\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{y})^2}{\sigma_y^2 \boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{a}_i(\boldsymbol{x})^T + \sigma_x^2 \boldsymbol{b}_i(\boldsymbol{y})\boldsymbol{b}_i(\boldsymbol{y})^T} \leq T_{\alpha,r} \tag{2.426}$$

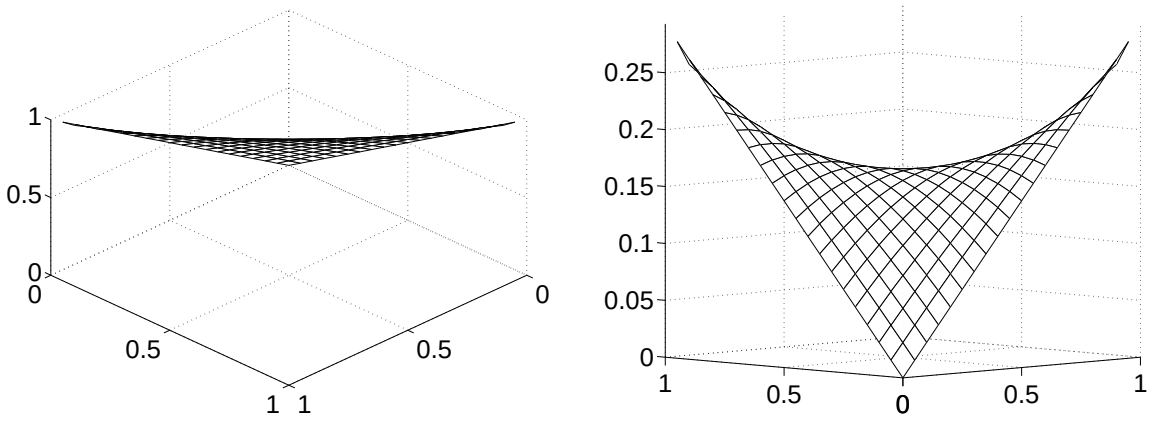

Figure 2.46: *Left:* The function $f(\delta_x, \delta_y) = \delta_x\sqrt{1-\delta_y^2}+\delta_y\sqrt{1-\delta_x^2}$ on the interval $\delta_x+\delta_y \leq 1$. Observe, how similar it looks to the function of just the sum of the two arguments $\delta_x + \delta_y$ on this interval. However, it is below the surface $\delta_x + \delta_y$, so that a factor of $\sqrt{2}$ has to be introduced. *Right:* The function $f(\delta_x, \delta_y) = \delta_x\sqrt{1-\delta_y^2} + \delta_y\sqrt{1-\delta_x^2} - \frac{1}{\sqrt{2}}(\delta_x + \delta_y)$, which reflects the approximation error introduced by the factor $\sqrt{2}$, on the interval $\delta_x + \delta_y \leq 1$. Observe that it is non-negative on this interval and that the largest approximation error occurs, if the values of δ_x and δ_y differ significantly.

$$\Leftrightarrow \quad (\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{y})^2 \leq T_{\alpha,r}(\sigma_y^2\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{a}_i(\boldsymbol{x})^T + \sigma_x^2\boldsymbol{b}_i(\boldsymbol{y})\boldsymbol{b}_i(\boldsymbol{y})^T) \tag{2.427}$$

$$\Leftrightarrow \quad |\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{y}| \leq \sqrt{T_{\alpha,r}(\sigma_y^2\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{a}_i(\boldsymbol{x})^T + \sigma_x^2\boldsymbol{b}_i(\boldsymbol{y})\boldsymbol{b}_i(\boldsymbol{y})^T)} \tag{2.428}$$

$$\Leftrightarrow \quad \frac{|\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{y}|}{|\boldsymbol{a}_i(\boldsymbol{x})||\boldsymbol{y}|} \leq \sqrt{T_{\alpha,r}\left(\frac{\sigma_y^2\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{a}_i(\boldsymbol{x})^T}{|\boldsymbol{a}_i(\boldsymbol{x})|^2|\boldsymbol{y}|^2} + \frac{\sigma_x^2\boldsymbol{b}_i(\boldsymbol{y})\boldsymbol{b}_i(\boldsymbol{y})^T}{|\boldsymbol{a}_i(\boldsymbol{x})|^2|\boldsymbol{y}|^2}\right)} \tag{2.429}$$

$$\Leftrightarrow \quad \frac{|\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{y}|}{|\boldsymbol{a}_i(\boldsymbol{x})||\boldsymbol{y}|} \leq \sqrt{T_{\alpha,r}\left(\frac{\sigma_y^2}{|\boldsymbol{y}|^2} + \frac{\sigma_x^2\boldsymbol{b}_i(\boldsymbol{y})\boldsymbol{b}_i(\boldsymbol{y})^T}{|\boldsymbol{a}_i(\boldsymbol{x})|^2|\boldsymbol{y}|^2}\right)} \tag{2.430}$$

Applying the same reasoning to equation (2.411) yields the equivalent signed condition

$$\frac{\boldsymbol{a}(\boldsymbol{x})\boldsymbol{y}}{\sqrt{\sigma_y^2\boldsymbol{a}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x})^T + \sigma_x^2\boldsymbol{b}(\boldsymbol{y})\mathsf{C}_{xx}\boldsymbol{b}(\boldsymbol{y})^T}} \leq \sqrt{T_{\alpha,1}} \tag{2.431}$$

$$\Leftrightarrow \quad \frac{\boldsymbol{a}(\boldsymbol{x})\boldsymbol{y}}{|\boldsymbol{a}(\boldsymbol{x})||\boldsymbol{y}|} \leq \sqrt{T_{\alpha,r}\left(\frac{\sigma_y^2}{|\boldsymbol{y}|^2} + \frac{\sigma_x^2\boldsymbol{b}(\boldsymbol{y})\boldsymbol{b}(\boldsymbol{y})^T}{|\boldsymbol{a}(\boldsymbol{x})|^2|\boldsymbol{y}|^2}\right)} \tag{2.432}$$

which differs only by the missing absolute value on the left hand side of the equation.

This conditions are further relaxed using the triangle inequality and the assumption that the inequality

$$\frac{|\boldsymbol{b}_i(\boldsymbol{y})|}{|\boldsymbol{y}|} \leq 1 \tag{2.433}$$

holds. This assumption is true for all relations discussed in the previous section. The final necessary conditions, which are easily interpretable as will become clearer in the following, are now given by (see also figure 2.46, left)

$$\frac{|\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{y}|}{|\boldsymbol{a}_i(\boldsymbol{x})||\boldsymbol{y}|} \quad \leq \quad \sqrt{T_{\alpha,r}}\frac{\sigma_y}{|\boldsymbol{y}|} + \sqrt{T_{\alpha,r}}\frac{\sigma_x}{|\boldsymbol{a}_i(\boldsymbol{x})|}\frac{|\boldsymbol{b}_i(\boldsymbol{y})|}{|\boldsymbol{y}|} \tag{2.434}$$

$$\leq \sqrt{T_{\alpha,r}}\frac{\sigma_y}{|\boldsymbol{y}|} + \sqrt{T_{\alpha,r}}\frac{\sigma_x}{|\boldsymbol{a}_i(\boldsymbol{x})|} \tag{2.435}$$

$$\leq \begin{cases} \delta_x\sqrt{1-\delta_y^2} + \delta_y\sqrt{1-\delta_x^2} & \text{if } \delta_x^2+\delta_y^2 \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{2.436}$$

and

$$\frac{\boldsymbol{a}(\boldsymbol{x})\boldsymbol{y}}{|\boldsymbol{a}(\boldsymbol{x})||\boldsymbol{y}|} \leq \begin{cases} \delta_x\sqrt{1-\delta_y^2} + \delta_y\sqrt{1-\delta_x^2} & \text{if } \delta_x^2+\delta_y^2 \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{2.437}$$

with the substitutions (see below)

$$\delta_x = \sqrt{2T_{\alpha,r}}\frac{\sigma_x}{|\boldsymbol{a}_i(\boldsymbol{x})|} \tag{2.438}$$

and

$$\delta_y = \sqrt{2T_{\alpha,r}}\frac{\sigma_y}{|\boldsymbol{y}|} \tag{2.439}$$

To see the inequalities (2.436) and (2.437), one first notes that the case for $\delta_x^2+\delta_y^2 > 1$ is trivial, as there is a cosine on the left hand side of the equation, which cannot be larger than one. For the case of $\delta_x^2+\delta_y^2 \leq 1$ one looks at the difference function (see figure 2.46, right)

$$\delta_x\sqrt{1-\delta_y^2} + \delta_y\sqrt{1-\delta_x^2} - \left(\sqrt{T_{\alpha,r}}\frac{\sigma_y}{|\boldsymbol{y}|} + \sqrt{T_{\alpha,r}}\frac{\sigma_x}{|\boldsymbol{a}_i(\boldsymbol{x})|}\right) \tag{2.440}$$

$$= \delta_x\sqrt{1-\delta_y^2} + \delta_y\sqrt{1-\delta_x^2} - \frac{1}{\sqrt{2}}(\delta_x+\delta_y) \tag{2.441}$$

$$= f(\delta_x,\delta_y) \tag{2.442}$$

and prooves that $f(\delta_x,\delta_y) \geq 0$ on the set

$$\{(\delta_x,\delta_y)|\delta_x>0,\delta_y>0,\delta_x^2+\delta_y^2\leq 1\} \tag{2.443}$$

Therefore one computes the derivative

$$\frac{\partial f(\delta_x,\delta_y)}{\partial \delta_x} = \sqrt{1-{\delta_y}^2} - \frac{\delta_x\delta_y}{\sqrt{1-{\delta_x}^2}} - \frac{1}{\sqrt{2}} \tag{2.444}$$

Setting this equal to zero and solving for δ_y yields the two solutions

$$\overline{\delta_y} = -\frac{1}{\sqrt{2}}\frac{\sqrt{(1-\delta_x)(\delta_x+1)}\left({\delta_x}^2 \pm \sqrt{{\delta_x}^2+{\delta_x}^4}\right)}{\delta_x} \tag{2.445}$$

One easily verifies using the triangle inequality that only the '$-$'-solution is positive for $0 \leq \delta_x \leq 1$. Substituting this solution back into equation (2.441) and setting its derivative equal to zero

$$\frac{\partial f(\delta_x,\overline{\delta_y})}{\partial \delta_x} = 0 \tag{2.446}$$

one obtains the two solutions

$$\overline{\delta_x} = \pm\frac{1}{4}\sqrt{7-\sqrt{17}} \tag{2.447}$$

Obviously only the '+'-solution is positive. Substituting this solution back into equation (2.445) yields the position of the only local extremum of (2.441) on the set defined in equation (2.443). Evaluating (2.441) at this position, one easily verifies that it is greater than zero

$$f\left(\frac{1}{4}\sqrt{7-\sqrt{17}}, \frac{1}{4}\sqrt{7-\sqrt{17}}\right) \approx 0.17 > 0 \tag{2.448}$$

It remains to check the boundary of the set defined in equation (2.443). For the two boundaries

$$f(\delta_x, 0) = \left(1 - \sqrt{\frac{1}{2}}\right)\delta_x \tag{2.449}$$

and

$$f(0, \delta_y) = \left(1 - \sqrt{\frac{1}{2}}\right)\delta_y \tag{2.450}$$

one can immediately see that the function is greater than zero on the intervals $0 \leq \delta_x \leq 1$ and $0 \leq \delta_y \leq 1$. Evaluating the function on the remaining third boundary yields

$$f\left(\delta_x, \sqrt{1-\delta_x^2}\right) = 1 - \sqrt{\frac{1}{2}}\left(\delta_x + \sqrt{1-\delta_x^2}\right) \tag{2.451}$$

which is greater than zero for the two values $\delta_x = 0$ and $\delta_x = 1$ as has already been noted above. Its derivative is

$$\frac{\partial f\left(\delta_x, \sqrt{1-\delta_x^2}\right)}{\partial \delta_x} = -\sqrt{\frac{1}{2}}\left(1 - \frac{\delta_x}{\sqrt{1-\delta_x^2}}\right) \tag{2.452}$$

Setting it equal to zero and solving for δ_x yields the position of its extremum

$$\overline{\delta_x} = \sqrt{\frac{1}{2}} \tag{2.453}$$

Substituting this back into equation (2.451) finally yields the value to be

$$f\left(\overline{\delta_x}, \sqrt{1-\overline{\delta_x}^2}\right) = 0 \tag{2.454}$$

which finally proofs that $f(\delta_x, \delta_y) \geq 0$ on the set defined in equation (2.443) and that equality is actually achieved at this position justifying the choice of the substitution δ_x and δ_y.

The final necessary condition can now be reformulated in order to facilitate an easy interpretation. First note that in case $\delta_x^2 + \delta_y^2 \leq 1$ the following identity holds (cf. [Bronstein *et al.*, 2001, p.87])

$$\begin{aligned}
& \delta_x\sqrt{1-\delta_y^2} + \delta_y\sqrt{1-\delta_x^2} && (2.455)\\
= \;& \sin\arcsin\left(\delta_x\sqrt{1-\delta_y^2} + \delta_y\sqrt{1-\delta_x^2}\right) && (2.456)\\
= \;& \sin\left(\arcsin\delta_x + \arcsin\delta_y\right) && (2.457)\\
= \;& \cos\left(\frac{\pi}{2} - \left(\arccos\delta_x + \arccos\delta_y\right)\right) && (2.458)
\end{aligned}$$

Hence, equation (2.436) is equivalent to

$$\frac{|\boldsymbol{a}_i(\boldsymbol{x})\boldsymbol{y}|}{|\boldsymbol{a}_i(\boldsymbol{x})||\boldsymbol{y}|} \leq \begin{cases} \cos\left(\frac{\pi}{2} - \left(\arccos\delta_x + \arccos\delta_y\right)\right) & \text{if } \delta_x^2 + \delta_y^2 \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{2.459}$$

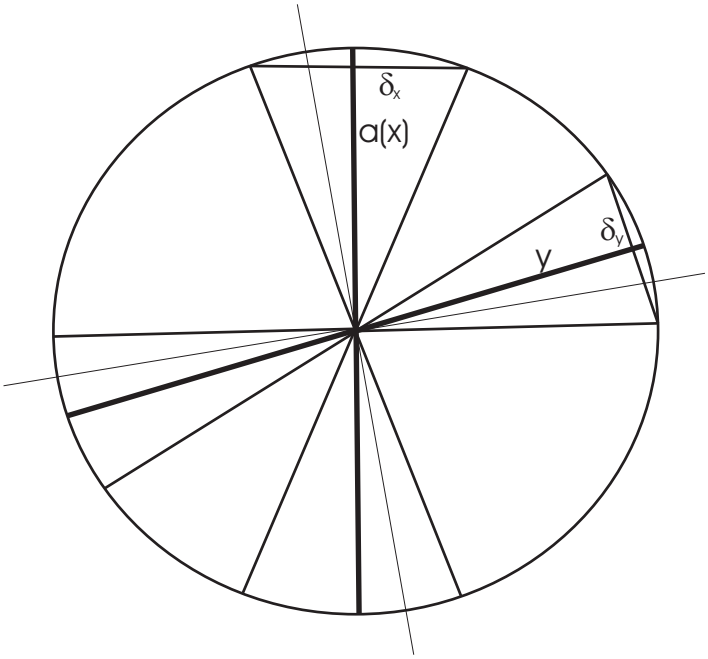

Figure 2.47: The necessary condition can be interpreted geometrically as follows: inside the cone with axis $\boldsymbol{y}$ and radius δ_y on the unit sphere there exists a direction vector, which is perpendicular to another direction vector, existing inside the cone with axis $\boldsymbol{a}(\boldsymbol{x})$ and radius δ_x on the unit sphere.

which can be interpreted as follows (see figure 2.47): Whenever a bilinear test (cf. equation (2.270)) gives no reason to assume that a given relation between two entities $\boldsymbol{x}$ and $\boldsymbol{y}$ does not hold, then for all i the angle between the vectors $\boldsymbol{a}_i(\boldsymbol{x})$ and $\boldsymbol{y}$ differs at most the amount $\arccos\delta_x + \arccos\delta_y$ from the right angle.

If on the other hand the contrary is true and the angle differs more than this amount, the bilinear test (cf. equation (2.270)) will be definitely rejected. This latter property is the key to the data structure presented here, as the values δ_y can be computed in advance and, as will be shown in the next section, can also be combined for multiple entities.

In the signed case (cf. (2.273)) equation (2.437) is equivalent to

$$\frac{\boldsymbol{a}(\boldsymbol{x})\boldsymbol{y}}{|\boldsymbol{a}(\boldsymbol{x})||\boldsymbol{y}|} \leq \begin{cases} \cos\left(\frac{\pi}{2} - (\arccos\delta_x + \arccos\delta_y)\right) & \text{if } \delta_x^2 + \delta_y^2 \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{2.460}$$

which can be interpreted as follows: whenever there is no reason to assume that a given signed bilinear relation between two entities $\boldsymbol{x}$ and $\boldsymbol{y}$ does not hold, then either the sign of the oriented angle is negative or the angle between the vectors $\boldsymbol{a}(\boldsymbol{x})$ and $\boldsymbol{y}$ differs at most the amount $\arccos\delta_x + \arccos\delta_y$ from the right angle. Again the negation of this statement is the key to the data structure.

The pairs $[\boldsymbol{y}, \delta_y]$ will be called keys in the following, as they allow to check the necessary conditions for every relation. As noted above, the keys have a geometric interpretation: they represent cones having the axis $\boldsymbol{y}$ and the radius δ_y at the intersection with the unit sphere.

2.3.2 Combining necessary conditions

The next step is combining the necessary conditions for two entities into one single necessary condition. That means given two keys $[\boldsymbol{y}_1, \delta_{y_1}]$ and $[\boldsymbol{y}_2, \delta_{y_2}]$ that allow to check the two conditions

$$\frac{|\boldsymbol{a}(\boldsymbol{x})\boldsymbol{y}_1|}{|\boldsymbol{a}(\boldsymbol{x})||\boldsymbol{y}_1|} \leq \begin{cases} \cos\left(\frac{\pi}{2} - (\arccos\delta_x + \arccos\delta_{y_1})\right) & \text{if } \delta_x^2 + \delta_{y_1}^2 \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{2.461}$$

and

$$\frac{|\boldsymbol{a}(\boldsymbol{x})\boldsymbol{y}_2|}{|\boldsymbol{a}(\boldsymbol{x})||\boldsymbol{y}_2|} \leq \begin{cases} \cos\left(\frac{\pi}{2} - (\arccos\delta_x + \arccos\delta_{y_2})\right) & \text{if } \delta_x^2 + \delta_{y_2}^2 \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{2.462}$$

as shown in the previous section, a new key $[\boldsymbol{y}', \delta_y']$ has to be computed, so that the combined necessary condition

$$\frac{|\boldsymbol{a}(\boldsymbol{x})\boldsymbol{y}'|}{|\boldsymbol{a}(\boldsymbol{x})||\boldsymbol{y}'|} \leq \begin{cases} \cos\left(\frac{\pi}{2} - (\arccos\delta_x + \arccos\delta_y')\right) & \text{if } \delta_x^2 + \delta_y'^2 \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{2.463}$$

holds, whenever any of the two previous conditions holds. If this is possible, a tree-shaped data structure for efficient performing large amount of bi-linear tests can be devised. In the following it will be shown, how such a super-key can be computed.

The geometric interpretability is extremely helpful to devise an algorithm. First observe that this problem is by construction inherently two-dimensional in the plane spanned by the two vectors $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. The projection into this plane is depicted in figure 2.48. All directions in that plane must fulfill

$$\boldsymbol{e}_1 = (1 - \lambda_1)\boldsymbol{y}_1 + \lambda_1\boldsymbol{y}_2 \tag{2.464}$$

where points on the two envelope lines fulfill the additional constraint given by the Pythagoras theorem that

$$\left(\frac{\boldsymbol{y}_1^T\boldsymbol{e}_1}{|\boldsymbol{y}_1||\boldsymbol{e}_1|}\right)^2 + \delta_{y_1}^2 = 1 \tag{2.465}$$

Substituting $\boldsymbol{e}_1$ into this equation yields

$$\begin{aligned} 1 - \delta_{y_1}^2 &= \left(\frac{(1-\lambda_1)\boldsymbol{y}_1^T\boldsymbol{y}_1 + \lambda_1\boldsymbol{y}_1^T\boldsymbol{y}_2}{|\boldsymbol{y}_1||(1-\lambda_1)\boldsymbol{y}_1 + \lambda_1\boldsymbol{y}_2|}\right)^2 && (2.466) \\ &= \frac{((1-\lambda_1)\boldsymbol{y}_1^T\boldsymbol{y}_1 + \lambda_1\boldsymbol{y}_1^T\boldsymbol{y}_2)^2}{\boldsymbol{y}_1^T\boldsymbol{y}_1((1-\lambda_1)^2\boldsymbol{y}_1^T\boldsymbol{y}_1 + 2(\lambda_1 - \lambda_1^2)\boldsymbol{y}_1^T\boldsymbol{y}_2 + \lambda_1^2\boldsymbol{y}_2^T\boldsymbol{y}_2)} && (2.467) \\ &= \frac{((1-\lambda_1)a + \lambda_1 c)^2}{a((1-\lambda_1)^2 a + 2(\lambda_1 - \lambda_1^2)c + \lambda_1^2 b)} && (2.468) \end{aligned}$$

where the abreviations $a = \boldsymbol{y}_1^T\boldsymbol{y}_1$, $b = \boldsymbol{y}_2^T\boldsymbol{y}_2$ and $c = \boldsymbol{y}_1^T\boldsymbol{y}_2$ are introduced to allow a more compact notation. Solving this equation for λ_1 yields the two solutions

$$\overline{\lambda_1}^{\pm} = \frac{\left(\delta_{y_1}a - \delta_{y_1}c \pm \sqrt{\left(1 - \delta_{y_1}^2\right)(ab - c^2)}\right)a\delta_{y_1}}{c^2 - ab + \delta_{y_1}^2 a^2 - 2\delta_{y_1}^2 ac + \delta_{y_1}^2 ab} \tag{2.469}$$

Substituting this back into equation (2.464) yields two points on the two envelope lines

$$\overline{\boldsymbol{e}_1}^{\pm} = (1 - \overline{\lambda_1}^{\pm})\boldsymbol{y}_1 + \overline{\lambda_1}^{\pm}\boldsymbol{y}_2 \tag{2.470}$$

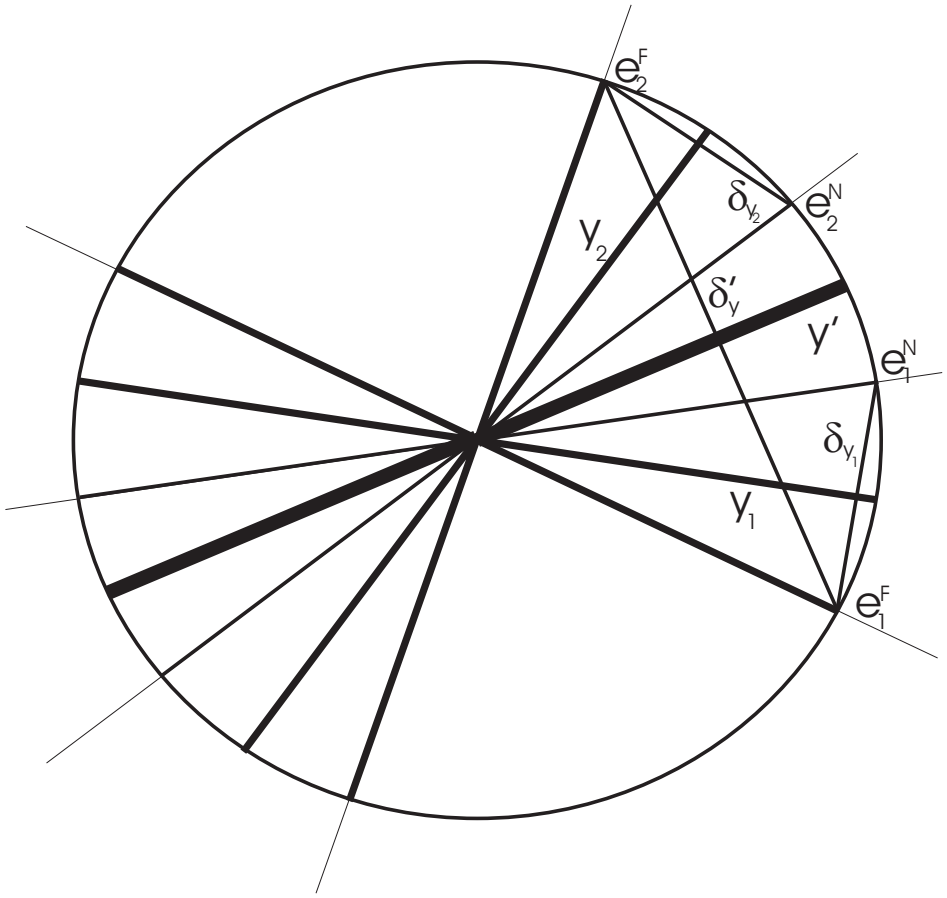

Figure 2.48: The two cones represented by the keys $[\boldsymbol{y}_1, \delta_{y_1}]$ and $[\boldsymbol{y}_2, \delta_{y_2}]$ are combined into a super-key $[\boldsymbol{y}', \delta'_y]$ by finding the union cone that is delimited by the two lines $\mathbf{e}_1^F$ and $\mathbf{e}_2^F$, which have the largest angle with each other among the delimiting lines.

of the first entity.

The same reasoning is possible for the envelope lines of the second entity

$$\boldsymbol{e}_2 = (1-\lambda_2)\boldsymbol{y}_2 + \lambda_2 \boldsymbol{y}_1 \tag{2.471}$$

for which the condition

$$\left(\frac{\boldsymbol{y}_2^T \boldsymbol{e}_2}{|\boldsymbol{y}_2||\boldsymbol{e}_2|}\right)^2 + \delta_{y_2}^2 = 1 \tag{2.472}$$

must be fulfilled. Substituting $\boldsymbol{e}_2$ into this equation now yields

$$\begin{aligned} 1-\delta_{y_2}^2 &= \left(\frac{(1-\lambda_2)\boldsymbol{y}_2^T\boldsymbol{y}_2 + \lambda_2 \boldsymbol{y}_1^T\boldsymbol{y}_2}{|\boldsymbol{y}_2||(1-\lambda_2)\boldsymbol{y}_2 + \lambda_2\boldsymbol{y}_1|}\right)^2 && (2.473)\\ &= \frac{((1-\lambda_2)\boldsymbol{y}_2^T\boldsymbol{y}_2 + \lambda_2\boldsymbol{y}_1^T\boldsymbol{y}_2)^2}{\boldsymbol{y}_2^T\boldsymbol{y}_2((1-\lambda_2)^2\boldsymbol{y}_2^T\boldsymbol{y}_2 + 2(\lambda_2-\lambda_2^2)\boldsymbol{y}_1^T\boldsymbol{y}_2 + \lambda_2^2\boldsymbol{y}_1^T\boldsymbol{y}_1)} && (2.474)\\ &= \frac{((1-\lambda_2)b + \lambda_2 c)^2}{b((1-\lambda_2)^2 b + 2(\lambda_2-\lambda_2^2)c + \lambda_2^2 a)} && (2.475) \end{aligned}$$

which has the two solutions

$$\overline{\lambda_2}^{\pm} = \frac{\left(\delta_{y_2}b - \delta_{y_2}c \pm \sqrt{\left(1-\delta_{y_2}^2\right)(ab-c^2)}\right) b\delta_{y_2}}{c^2 - ab + \delta_{y_2}^2 b^2 - 2\delta_{y_2}^2 bc + \delta_{y_2}^2 ab} \tag{2.476}$$

that are back-substituted into equation (2.471) yielding two points on the envelope lines of the second entity

$$\overline{\boldsymbol{e}_2}^{\pm} = (1-\overline{\lambda_2}^{\pm})\boldsymbol{y}_2 + \overline{\lambda_2}^{\pm}\boldsymbol{y}_1 \tag{2.477}$$

Now the two oriented envelope lines including the largest angle have to be selected. Therefore first the orientations have to be aligned with the entities. Also applying a spherical normalization yields the oriented envelope points

$$\tilde{\boldsymbol{e}_1}^{\pm} = \mathrm{sign}(\boldsymbol{y}_1^T\overline{\boldsymbol{e}_1}^{\pm})\frac{\overline{\boldsymbol{e}_1}^{\pm}}{|\overline{\boldsymbol{e}_1}^{\pm}|} \tag{2.478}$$

and

$$\tilde{\boldsymbol{e}_2}^{\pm} = \mathrm{sign}(\boldsymbol{y}_2^T\overline{\boldsymbol{e}_2}^{\pm})\frac{\overline{\boldsymbol{e}_2}^{\pm}}{|\overline{\boldsymbol{e}_2}^{\pm}|} \tag{2.479}$$

The next step is to distinguish for each entity between the envelope line that is on the same side as the second entity and the envelope line that is on the other side than the second entity. This can be done by checking the signs of the scalar products

$$\boldsymbol{e}_1^N = \begin{cases} \tilde{\boldsymbol{e}_1}^+ & \text{if } \boldsymbol{y}_2^T\tilde{\boldsymbol{e}_1}^+ > \boldsymbol{y}_2^T\tilde{\boldsymbol{e}_1}^- \\ \tilde{\boldsymbol{e}_1}^- & \text{otherwise} \end{cases} \tag{2.480}$$

$$\boldsymbol{e}_1^F = \begin{cases} \tilde{\boldsymbol{e}_1}^+ & \text{if } \boldsymbol{y}_2^T\tilde{\boldsymbol{e}_1}^+ < \boldsymbol{y}_2^T\tilde{\boldsymbol{e}_1}^- \\ \tilde{\boldsymbol{e}_1}^- & \text{otherwise} \end{cases} \tag{2.481}$$

$$\boldsymbol{e}_2^N = \begin{cases} \tilde{\boldsymbol{e}_2}^+ & \text{if } \boldsymbol{y}_1^T\tilde{\boldsymbol{e}_2}^+ > \boldsymbol{y}_1^T\tilde{\boldsymbol{e}_2}^- \\ \tilde{\boldsymbol{e}_2}^- & \text{otherwise} \end{cases} \tag{2.482}$$

$$\boldsymbol{e}_2^F = \begin{cases} \tilde{\boldsymbol{e}_2}^+ & \text{if } \boldsymbol{y}_1^T \tilde{\boldsymbol{e}_2}^+ < \boldsymbol{y}_1^T \tilde{\boldsymbol{e}_2}^- \\ \tilde{\boldsymbol{e}_2}^- & \text{otherwise} \end{cases} \tag{2.483}$$

Finally, as the two envelopes need not be disjoint, one obtains the two enclosing envelope lines, i.e. those with the largest opening angle, again from checking signs of scalar products

$$\boldsymbol{m} = \begin{cases} \boldsymbol{e}_1^F & \text{if } \boldsymbol{y}_2^T \boldsymbol{e}_1^F < \boldsymbol{y}_2^T \boldsymbol{y}_2^N \\ \boldsymbol{e}_2^N & \text{otherwise} \end{cases} \tag{2.484}$$

$$\boldsymbol{n} = \begin{cases} \boldsymbol{e}_2^F & \text{if } \boldsymbol{y}_1^T \boldsymbol{e}_2^F < \boldsymbol{y}_1^T \boldsymbol{y}_1^N \\ \boldsymbol{e}_1^N & \text{otherwise} \end{cases} \tag{2.485}$$

The new key $[\boldsymbol{y}', \delta'_y]$ is now in the middle between the two envelope lines

$$\boldsymbol{y}' = \frac{\boldsymbol{m} + \boldsymbol{n}}{|\boldsymbol{m} + \boldsymbol{n}|} \tag{2.486}$$

with the opening angle being computable using Pythagoras theorem as

$$\delta'_y = \begin{cases} \sqrt{1 - (\boldsymbol{m}^T \boldsymbol{n})^2} & \text{if } \boldsymbol{y}_1^T \boldsymbol{y}' > 0 \wedge \boldsymbol{y}_2^T \boldsymbol{y}' > 0 \\ 1 & \text{otherwise} \end{cases} \tag{2.487}$$

where the two cases must be distinguished to deal with angles greater than $90°$.

Note that, as the combined necessary condition has the same structure as the two single necessary conditions, it is possible to combine more than two keys successively. In the following a tree data structure based on those pre-computable keys representing necessary conditions for relations will be derived.

2.3.3 The tree data structure

Having defined necessary conditions for bilinear tests based on pre-computable keys for each entity and a method for combining such keys into super-keys, which allow to check the necessary condition for multiple entities at once, an R-tree like data structure (cf. [Guttman, 1984] and [Bayer and McCreight, 1972]) can be devised. This tree should have the following properties (see figure 2.49)

1. Every leaf node contains at least a pre-specified number M and at most $2M-1$ uncertain oriented projective entities together with the corresponding keys $[\boldsymbol{y}_i, \delta_{y_i}]$, which allow to check necessary conditions as described in section 2.3.1. If it is the root of the tree it may contain less elements.

2. Every inner node contains at least M and at most $2M - 1$ descendants together with the super-keys $[\boldsymbol{y}'_i, \delta'_{y_i}]$, such that all keys contained in the subtree are sufficient for the corresponding super-key as described in section 2.3.2.

Those properties ensure that the tree has logarithmic depth and does not degenerate, and therefore enable efficient access. In the following a query algorithm for such a tree data structure will be presented as well as an insertion and a deletion algorithm that preserve the properties described above by reorganizing the tree if required.

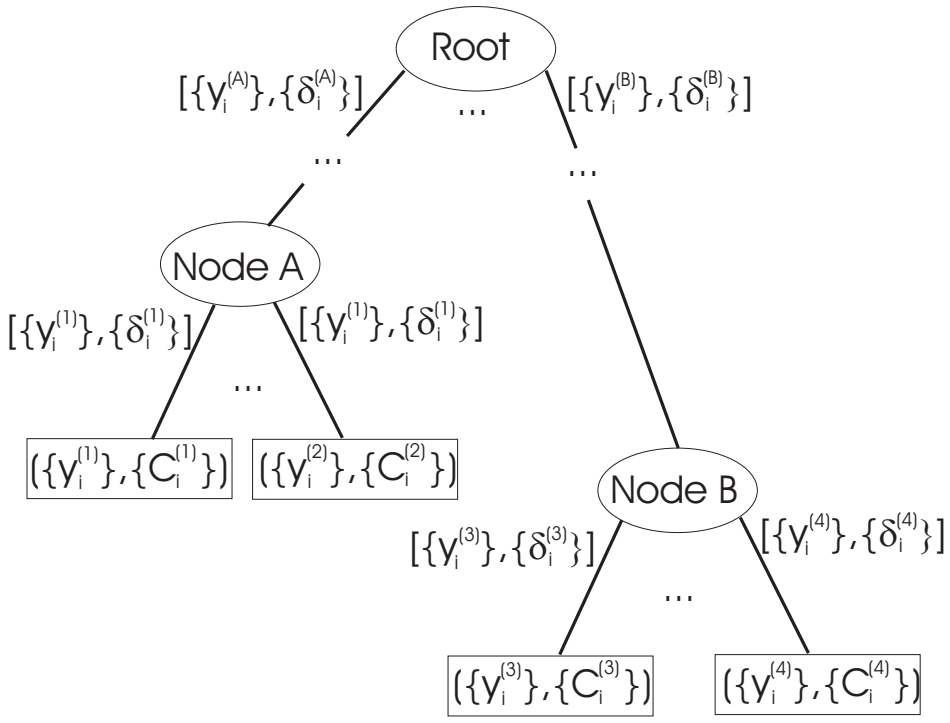

Figure 2.49: The tree structure contains an oriented projective entity in each leaf node. Each edge in this tree is label with a key, representing the cone in which all oriented projective sub-elements can be found.

Query

A query to such a data-structure is as follows: starting at the root for every descendant the necessary condition for the query is computed as has been derived in detail in section 2.3.1. If the condition is not fulfilled, then the corresponding subtree cannot contain an element that would fulfill the query. Therefore such subtrees are skipped and one descends only into subtrees, where the necessary condition is fulfilled. If the leafs of the tree are reached, the query is tested for each uncertain oriented projective element that is contained in the leaf node.

As the depth of the tree is logarithmic, the time complexity depends on how the query relation prunes the sub-trees so that a logarithmic running time is possible. Because the output size might be the whole set of stored elements, only linear time complexity can be ensured.

The details for the query are shown in algorithm 1.

Note that it might be more efficient even to test the necessary condition before line 4 of algorithm 1, if the evaluation of the relation is complex.

In line 11 of algorithm 1 it is required to check the condition, if some uncertain element fulfilling the query relation can possibly be found in the according subtree. The details for checking conditions are shown in algorithm 2.

Observe that the computations in lines 16, 18, 25 and 26 of algorithm 2 need only be performed once fore each query, as they depend only on the query relation and the query entity, which are known in advance and do not change during the query.

Algorithm 1 Query

Require: uncertain query element $(\{\mathbf{x}_i\}, \{\mathsf{C}_{x_i x_i}\})$
query relation $R(\{\mathbf{x}_i\}, \cdot)$
root of the tree $\mathcal{T}$
Ensure: set $\mathcal{S} = \{(\{\mathbf{y}_i\}, \{\mathsf{C}_{y_i y_i}\}) \mid R(\{\mathbf{x}_i\}, \{\mathbf{y}_i\}) \wedge (\{\mathbf{y}_i\}, \{\mathsf{C}_{y_i y_i}\}) \in \mathcal{T}\}$

```
S = ∅
if T is a leaf node then
  for each element ({y_i}, {C_{y_i y_i}}) of T do
    if R({x_i}, {y_i}) then
      S = S ∪ ({y_i}, {C_{y_i y_i}})
    end if
  end for
else
  for each descendant t of T do
    get key [{y'_i}, {δ'_{y_i}}]_t stored for descendant t
    if CheckCondition(({x_i}, {C_{x_i x_i}}), R({x_i}, ·), [{y'_i}, {δ'_{y_i}}]_t) then
      recursivly query subtree t resulting in the set S_t
      S = S ∪ S_t
    end if
  end for
end if
```

Insert

To insert a new element into the data-structure with logarithmic time complexity one proceeds as follows: starting at the root it is decided, into which descendant the element should be inserted and this process is iterated until a leaf is reached. The element is then inserted into this leaf node. There are three issues to address:

1. the decision rule for choosing an appropriate descendant has to be devised
2. the keys along the path have to be updated in order to guarantee that the inserted entity can be found
3. if the size of the leaf would exceed the maximum of $2M - 1$ after insertion, the tree has to be reorganized

For choosing a decision rule the geometric interpretation of the necessary conditions (2.459) and (2.460) is helpful. As noted before, any key $[\boldsymbol{y}, \delta_y]$ represents a cone with axis $\boldsymbol{y}$ and radius δ_y. The smaller the radius, the more descendants are pruned during a query. Therefore a reasonable choice as decision rule is to choose the descendant for insertion that requires the minimum increase of radius of the cone represented by its key.

This brings up the second issue of updating the key for the descendant, in which the element should be inserted. This has been derived in detail in section 2.3.2.

The third issue of reorganizing the tree is analogous to the reorganization of a B-tree (cf. [Bayer and McCreight, 1972]) by splitting up the overfull node and inserting it into its parent node. As the parent node can overflow as well, it might be necessary to iterate this process up the the root of the tree. As in the case of B-trees this process has at most logarithmic time complexity. Splitting a node brings up again the issues of choosing the two subsets and re-computing keys, which will be detailed below.

Algorithm 2 Check condition

Require: uncertain query element $(\{\mathbf{x}_i\}, \{\mathsf{C}_{x_i x_i}\})$
query relation $R(\{\mathbf{x}_i\}, \cdot)$
key $[\{\mathbf{y}_i\}, \{\delta_{y_i}\}]$

Ensure: **false**, if a sub-tree with the given key cannot contain an element fulfilling the query relation, and **true** otherwise

if $R(\{\mathbf{x}_i\}, \cdot) = R_1(\{\mathbf{x}_i\}, \cdot) \wedge R_2(\{\mathbf{x}_i\}, \cdot)$ **then**
  if CheckCondition$((\{\mathbf{x}_i\}, \{\mathsf{C}_{x_i x_i}\}), R_1(\{\mathbf{x}_i\}, \cdot), [\{\mathbf{y}_i\}, \{\delta_{y_i}\}])$ **then**
    if CheckCondition$((\{\mathbf{x}_i\}, \{\mathsf{C}_{x_i x_i}\}), R_2(\{\mathbf{x}_i\}, \cdot), [\{\mathbf{y}_i\}, \{\delta_{y_i}\}])$ **then**
      return true
    end if
  end if
  return false
else if $R(\{\mathbf{x}_i\}, \cdot) = R_1(\{\mathbf{x}_i\}, \cdot) \vee R_2(\{\mathbf{x}_i\}, \cdot)$ **then**
  if CheckCondition$((\{\mathbf{x}_i\}, \{\mathsf{C}_{x_i x_i}\}), R_1(\{\mathbf{x}_i\}, \cdot), [\{\mathbf{y}_i\}, \{\delta_{y_i}\}])$ **then**
    return true
  else if CheckCondition$((\{\mathbf{x}_i\}, \{\mathsf{C}_{x_i x_i}\}), R_2(\{\mathbf{x}_i\}, \cdot), [\{\mathbf{y}_i\}, \{\delta_{y_i}\}])$ **then**
    return true
  end if
  return false
else if $R(\{\mathbf{x}_i\}, \cdot) = \mathsf{A}(\mathbf{x}_i) \cdot \overset{!}{=} 0$ **then**
  compute σ_x from $\mathsf{C}_{x_i x_i}$ according to (2.407)
  for each row $\mathbf{a}_j$ of $\mathsf{A}(\mathbf{x}_i)$ **do**
    compute δ_x^j from $\mathbf{a}_j$ and σ_x according to (2.438)
    if condition (2.436) does not hold **then**
      return false
    end if
  end for
  return true
else if $R(\{\mathbf{x}_i\}, \cdot) = \mathbf{a}(\mathbf{x}_i) \cdot \overset{!}{\leq} 0$ **then**
  compute σ_x from $\mathsf{C}_{x_i x_i}$ according to (2.407)
  compute δ_x from $\mathbf{a}(\mathbf{x}_i)$ and σ_x according to (2.438)
  if condition (2.437) does not hold **then**
    return false
  end if
  return true
end if

The details for the insertion of elements are shown in algorithm 3.

Algorithm 3 Insert element

Require: compound uncertain oriented entitiy $(\{\mathbf{y}_i\}, \{\mathsf{C}_{y_i y_i}\})$ to be inserted
root of the tee $\mathcal{T}$
Ensure: tree $\mathcal{T}$ containing the element
1: **for** each base entity $(\mathbf{y}_i, \mathsf{C}_{y_i y_i})$ **do**
2: compute σ_{y_i} from $\mathsf{C}_{y_i y_i}$ according to (2.408)
3: compute δ_{y_i} from σ_{y_i} and $\boldsymbol{y}_i$ according to equation (2.439)
4: **end for**
5: construct key $[\{\mathbf{y}_i\}, \{\delta_{y_i}\}]$
6: **if** $\mathcal{T}$ is a leaf **then**
7: insert the element and its key into leaf $\mathcal{T}$
8: **if** size of $|\mathcal{T}| = 2M$ **then**
9: $\left(\mathcal{T}_1; [\{\boldsymbol{y}_i^{(1)}\}, \{\delta_{y_i}^{(1)}\}]; \mathcal{T}_2; [\{\boldsymbol{y}_i^{(2)}\}, \{\delta_{y_i}^{(2)}\}]\right) = \text{SplitNode}(\mathcal{T})$
10: **if** $\mathcal{T}$ is the root **then**
11: create new root $\mathcal{T}$
12: insert $\mathcal{T}_1$ and $\mathcal{T}_2$ into new root $\mathcal{T}$ using algorithm 4
13: **else**
14: remove $\mathcal{T}$ from its parent node
15: insert $\mathcal{T}_1$ and $\mathcal{T}_2$ into the previous parent of $\mathcal{T}$ using algorithm 4
16: **end if**
17: **end if**
18: **else**
19: **for** each descendant t of $\mathcal{T}$ **do**
20: get key $[\{\mathbf{y}_i^{(t)}\}, \{\delta_{y_i}^{(t)}\}]$ stored for descendant t
21: compute combined key $[\{\mathbf{y}_i'^{(t)}\}, \{\delta_{y_i}'^{(t)}\}]$ from $[\{\mathbf{y}_i^{(t)}\}, \{\delta_{y_i}^{(t)}\}]$ and $[\{\mathbf{y}_i\}, \{\delta_{y_i}\}]$ according to equations (2.486) and (2.487) as derived in section 2.3.2
22: **end for**
23: choose descendant $\hat{t} = \text{argmin}_t \max_i |\delta_{y_i}'^{(t)} - \delta_{y_i}^{(t)}|$ with minimum radius enlargement
24: set key for descendant $\hat{t}$ to $[\{\mathbf{y}_i'^{(\hat{t})}\}, \{\delta_{y_i}'^{(\hat{t})}\}]$
25: recursively insert the element into the descendant $\hat{t}$
26: **end if**

Note that the computation of the key in lines 1-5 of algorithm 3 needs only be performed once, as the key is only dependent on the entity itself and does not change during the recursion. In line 23 of algorithm 3 the L_∞-norm is minimized, though one could choose any other appropriate measure such as the L_1- or L_2-norm.

In lines 12 and 15 of algorithm 3 a whole subtree must be inserted into the data-structure. This is done completely analogous to the previous algorithm with the distinction that subtrees may also be inserted into inner nodes of the tree. The details for the insertion of trees are shown in algorithm 4.

Both insertion algorithms needed to split up an overflowing node at some point. Choosing an optimal bipartition of the descendants would require to check all of those bipartitions. As this is impractical even for small values of M, a heuristic similar to one used in R-tree (cf. [Guttman, 1984]) is applied. The idea is to first choose those two descendants that are farthest away from each other, i.e. those two descendants, where the radius of the cone represented by the key resulting from combining the keys of the two descendants is maximal. Starting

Algorithm 4 Insert Tree

Require: subtree $\mathcal{S}$ to be inserted
key $[\{\boldsymbol{y}_i\}, \{\delta_{y_i}\}]$ of the subtree
root of the tee $\mathcal{T}$
Ensure: tree $\mathcal{T}$ containing the subtree $\mathcal{S}$
1: insert sub-tree $\mathcal{S}$ and its key into $\mathcal{T}$
2: **if** size of $|\mathcal{T}| = 2M$ **then**
3: $\left(\mathcal{T}_1; [\{\boldsymbol{y}_i^{(1)}\}, \{\delta_{y_i}^{(1)}\}]; \mathcal{T}_2; [\{\boldsymbol{y}_i^{(2)}\}, \{\delta_{y_i}^{(2)}\}]\right) = \text{SplitNode}(\mathcal{T})$
4: **if** $\mathcal{T}$ is the root **then**
5: create new root $\mathcal{T}$
6: insert $\mathcal{T}_1$ and $\mathcal{T}_2$ into new root $\mathcal{T}$
7: **else**
8: remove $\mathcal{T}$ from its parent node
9: recursively insert $\mathcal{T}_1$ and $\mathcal{T}_2$ into the previous parent of $\mathcal{T}$
10: **end if**
11: **end if**

from those two descendants, each is augmented in turn with the closest descendant in the above sense until all descendants are distributed.

The details for the splitting of nodes are shown in algorithm 5.

As before, the choice of L_∞-norm in lines 6 and 16 of algorithm 5 could be replaced by other distance measures such as L_1- or L_2-norm.

Delete

Deletion of an element from the tree requires that the element inside the tree is somehow identified. This is possible using the query function, so that it will be assumed that the leaf containing the entity to be deleted has already been identified. There are two issues that have to be addressed: first, the deletion of an entity from a non-root node can result in a leaf containing less than M elements, so that the tree has to be reorganized; second, the keys on the path from the deleted element to the root may be re-computed in order to improve the query performance of the tree. As opposed to the insertion case, this re-computation of keys, i.e. lines 4-8 of algorithm 6, is not mandatory and may be skipped, depending on performance considerations.

The details for the deletion of elements are shown in algorithm 6.

Algorithm 5 Split node

Require: node of tree $\mathcal{T}$ having size $|\mathcal{T}| = 2M$

Ensure: tree $\mathcal{T}_1$ with key $[\{\boldsymbol{y}_i^{(1)}\}, \{\delta_{y_i}^{(1)}\}]$
tree $\mathcal{T}_2$ with key $[\{\boldsymbol{y}_i^{(2)}\}, \{\delta_{y_i}^{(2)}\}]$, with $\mathcal{T}_1 \cup \mathcal{T}_2 = \mathcal{T}$ and $|\mathcal{T}_1| = |\mathcal{T}_2| = M$

1: **for** each pair of descendants t_1 and t_2 of $\mathcal{T}$ **do**
2: get key $[\{\mathbf{y}_i^{(t_1)}\}, \{\delta_{y_i}^{(t_1)}\}]$ stored for descendant t_1
3: get key $[\{\mathbf{y}_i^{(t_2)}\}, \{\delta_{y_i}^{(t_2)}\}]$ stored for descendant t_2
4: compute combined key $[\{\mathbf{y}_i'^{(t_1,t_2)}\}, \{\delta_{y_i}'^{(t_1,t_2)}\}]$ from $[\{\mathbf{y}_i^{(t_1)}\}, \{\delta_{y_i}^{(t_1)}\}]$ and $[\{\mathbf{y}_i^{(t_2)}\}, \{\delta_{y_i}^{(t_2)}\}]$ according to equations (2.486) and (2.487) as derived in section 2.3.2
5: **end for**
6: choose pair $(\hat{t}_1, \hat{t}_2) = \operatorname{argmax}_{t_1,t_2} \max_i |\delta_{y_i}'^{(t_1,t_2)}|$ with maximum distance
7: remove chosen descendants, i.e. $\mathcal{T} = \mathcal{T} \backslash \{\hat{t}_1, \hat{t}_2\}$
8: set $\mathcal{T}_1 = \{\hat{t}_1\}$ and $[\{\boldsymbol{y}_i^{(1)}\}, \{\delta_{y_i}^{(1)}\}] = [\{\boldsymbol{y}_i^{(\hat{t}_1)}\}, \{\delta_{y_i}^{(\hat{t}_1)}\}]$
9: set $\mathcal{T}_2 = \{\hat{t}_2\}$ and $[\{\boldsymbol{y}_i^{(2)}\}, \{\delta_{y_i}^{(2)}\}] = [\{\boldsymbol{y}_i^{(\hat{t}_2)}\}, \{\delta_{y_i}^{(\hat{t}_2)}\}]$
10: **while** $\mathcal{T} \neq \emptyset$ **do**
11: **for** $\nu = 0..1$ **do**
12: **for** each remaining descendant t of $\mathcal{T}$ **do**
13: get key $[\{\mathbf{y}_i^{(t)}\}, \{\delta_{y_i}^{(t)}\}]$ stored for descendant t
14: compute combined key $[\{\mathbf{y}_i'^{(t)}\}, \{\delta_{y_i}'^{(t)}\}]$ from $[\{\mathbf{y}_i^{(t)}\}, \{\delta_{y_i}^{(t)}\}]$ and $[\{\mathbf{y}_i^{(\hat{t}_\nu)}\}, \{\delta_{y_i}^{(\hat{t}_\nu)}\}]$ according to equations (2.486) and (2.487) as derived in section 2.3.2
15: **end for**
16: choose descendant $\hat{t} = \operatorname{argmin}_t \max_i |\delta_{y_i}'^{(t)} - \delta_{y_i}^{(t)}|$ with minimum radius enlargement
17: add chosen descendant $\hat{t}$ to tree $\mathcal{T}_\nu$
18: update key $[\{\boldsymbol{y}_i^{(\nu)}\}, \{\delta_{y_i}^{(\nu)}\}] = [\{\mathbf{y}_i'^{(\hat{t})}\}, \{\delta_{y_i}'^{(\hat{t})}\}]$
19: remove descendant $\mathcal{T} = \mathcal{T} \backslash \{\hat{t}\}$
20: **end for**
21: **end while**

Algorithm 6 Delete Element

Require: leaf $\mathcal{L}$ containing the entity to be deleted
Ensure: tree $\mathcal{T}$ no longer containing the element
1: remove element and key from leaf $\mathcal{L}$
2: set $\mathcal{T} = \mathcal{L}$
3: **if** $\mathcal{L}$ is not the root **then**
4: **repeat**
5: compute combined key $[\{\mathbf{y}'_i\}, \{\delta_{y_i}\}]$ consecutively from all keys stored in $\mathcal{T}$ according to equations (2.486) and (2.487) as derived in section 2.3.2
6: replace the key stored for $\mathcal{T}$ in its parent by $[\{\mathbf{y}'_i\}, \{\delta_{y_i}\}]$
7: change $\mathcal{T}$ into its parent
8: **until** $\mathcal{T}$ is the root
9: **if** size of $|\mathcal{L}| < M$ **then**
10: remove $\mathcal{L}$ from its parent
11: insert remaining elements from $\mathcal{L}$ into its parent using algorithm 3
12: **end if**
13: **end if**

Up to now a data structure for storing uncertain oriented projective entities has been presented that enables testing relations. Although the depth of the tree is guaranteed to be logarithmic, the selectivity of the queries and the tightness of the necessary conditions plays a major role for its performance. In order to assess the performance of the presented data structure, an empirical performance evaluation will be given in the following.

2.3.4 Empirical performance analysis

To analyze the performance of the presented data structure, some experiments on synthetic data will be presented. First a set of uncertain oriented projective 2d line segments of random length varying from 0.05 to 0.5 lying uniformly distributed inside the unit box $[-0.5..0.5] \times [-0.5..0.5]$ were generated. The accuracy of the end-points was set to $\sigma = 0.001$. The running times for constructing the data structure for branching factors varying from $M = 2$ to $M = 64$ are plotted against the number of elements on the left hand side in figure 2.50. It can be seen that the best performance for constructing the data structure is achieved with the smallest possible branching factor $M = 2$.

Now another uncertain oriented projective 2d line segment is generated at random and all incident uncertain oriented projective line segments are retrieved from the data structure. The running times of this query operation for different branching factors varying from $M = 2$ to $M = 64$ are plotted against the number of elements contained in the data structure on the right hand side in figure 2.50. It can be seen that the branching factor has no significant impact on the performance for this query. Hence, a branching factor of $M = 2$ maximizes the overall performance, because in this case the construction is most efficient as has been observed before.

Next the performance of the data structure is compared to the naive method of sequentially testing all uncertain oriented projective 2d line segments contained in it. The running times of both competing methods are plotted against the number of elements on the left hand side in figure 2.51. It can be seen that a significant increase of query performance is gained from using the proposed data structure.

The next experiment demonstrates the performance for more complex elements. Therefore

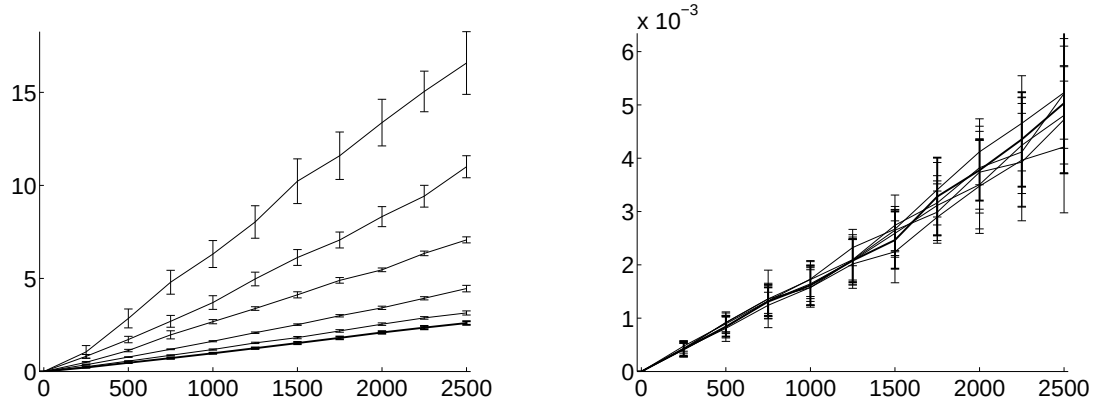

Figure 2.50: *Left:* Construction times (with standard deviations) for the data structure containing 2d line segments against their number for different branching factors. The thick line corresponds to a branching factor of $M = 2$. The other lines correspond with increasing running times to the branching factors $M = 4, 8, 16, 32$ and 64. *Right:* Query times (with standard deviations) for retrieving all incident 2d line segments against the number of 2d line segments contained in the data structure for different branching factors. The thick line corresponds to a branching factor of $M = 2$, the other lines to branching factors $M = 4, 8, 16, 32$ and 64. Observe that the branching factor has no significant influence on the running times in this case.

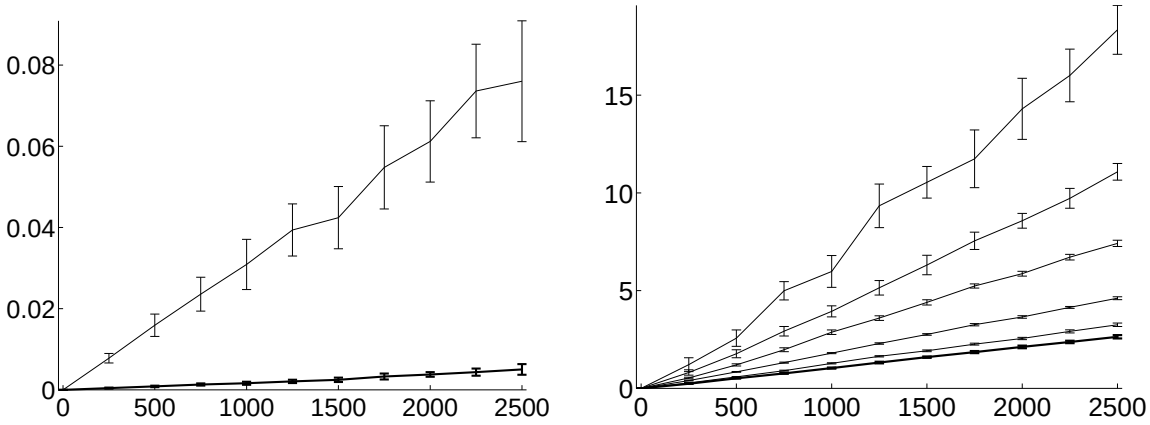

Figure 2.51: Left:Comparison of running times using the data structure with sequential testing not using the data structure. The thick line shows the query performance (with standard deviations) using the data structure with a branching factor of $M = 2$. The other line shows the query performance (with standard deviations), when all 2d line segments are tested for incidence sequentially. *Right:* Construction times (with standard deviations) for the data structure containing 3d line segments against their number for different branching factors. The thick line corresponds to a branching factor of $M = 2$. The other lines correspond with increasing running times to the branching factors $M = 4, 8, 16, 32$ and 64.

uncertain oriented projective 3d line segments were generated at random using the same parameters as before in the 2d case. That means the length of the line segments was allowed to vary between 0.05 and 0.5 and the positions were generated inside the unit box $[-0.5..0.5] \times [-0.5..0.5] \times [-0.5..0.5]$. The end-point accuracy was set to $\sigma = 0.001$. The construction times for different branching factors $M = 2, 4, 8, 16, 32$ and 64 are plotted against the number of elements on the right hand side in figure 2.51. It can be seen again that the construction performance is optimal for the minimal branching factor $M = 2$.

Again another uncertain oriented projective 3d line segment was generate using the same parameters as before. All uncertain oriented projective 3d line segments were retrieved from the data structure that are incident to the generated test segment. The running times of this query for different branching factors are plotted against the size of the data structure on the left hand side in figure 2.52. It can be seen that unlike the 2d case the performance is better for branching factors $M > 2$. Hence, for complex queries on higher dimensional entities the branching factor must be chosen, such that the construction time for the data structure is traded against the query time based on the expected number of queries.

On the right hand side in figure 2.52 the performance of the incident query on the data structure with a branching factor $M = 2$ is compared to sequential testing all elements. It can be seen that like the 2d case the performance is significantly increased, if the proposed data structure is used.

The final experiment demonstrates the influence of the accuracy of the elements contained in the data structure on the performance. The uncertain oriented projective 2d line segments from the first experiment are now stored in the data structure with end-point accuracies varying from $\sigma = 0.0001$ to $\sigma = 0.1$. On the left hand side in figure 2.53 the construction times for different end-point accuracies are plotted against the number of elements. It can be seen that the construction performance increases for the large value of $\sigma = 0.1$, because the keys do not need to be computed for each node any more. Unfortunately, those keys do

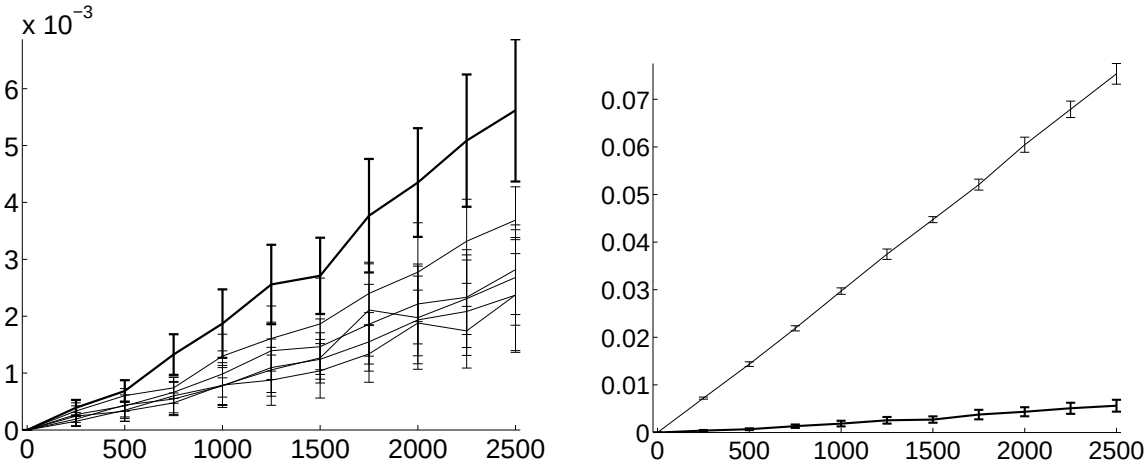

Figure 2.52: *Left:* Query times (with standard deviations) for retrieving all incident 3d line segments against the number of 3d line segments contained in the data structure for different branching factors. The thick line corresponds to a branching factor of $M = 2$, the other lines to branching factors $M = 4, 8, 16, 32$ and 64. *Right:* Comparison of running times using the data structure with sequential testing not using the data structure. The thick line shows the query performance (with standard deviations) using the data structure with a branching factor of $M = 2$. The other line shows the query performance (with standard deviations), when all 3d line segments are tested for incidence sequentially.

not carry any information either, which will be seen next.

Again all incident uncertain oriented projective 2d line segments are retrieved from the data structure for different end-point accuracies. The test entities are given the same end-point accuracy than the elements contained in the data structure. The running times for those queries together with the running time for sequentially testing all elements are plotted against the number of elements on the right hand side in figure 2.53. It can be seen that the performance decreases with the accuracies of the elements contained in the data structure. This is caused by two factors, which are also well-known from classical R-trees: first the result set becomes larger, so that effectively all elements have to be tested and second the keys in the upper nodes of the tree become less informative. Hence, if all elements potentially belong to the result set, then the overhead for the data structure decreases the performance of the query.

As a conclusion of the empirical analysis the presented data structure should be used, if a large amount of relations must be checked, because only in this case the construction times of the data structure justify its application. Furthermore it has been seen that the accuracy of the data to be tested must be reasonable good, so that the selectivity of the queries and the tightness of the necessary conditions yields an improvement against simple sequential testing. Both of those conditions are fulfilled for the application presented in chapter 4, so that significant runtime improvements have been achieved there.

This chapter discussed the representation of entities and efficient testing of relations in the uncertain oriented projective framework. In the following this two building blocks will be used to devise grouping algorithms for such entities that enable a wide range of applications.

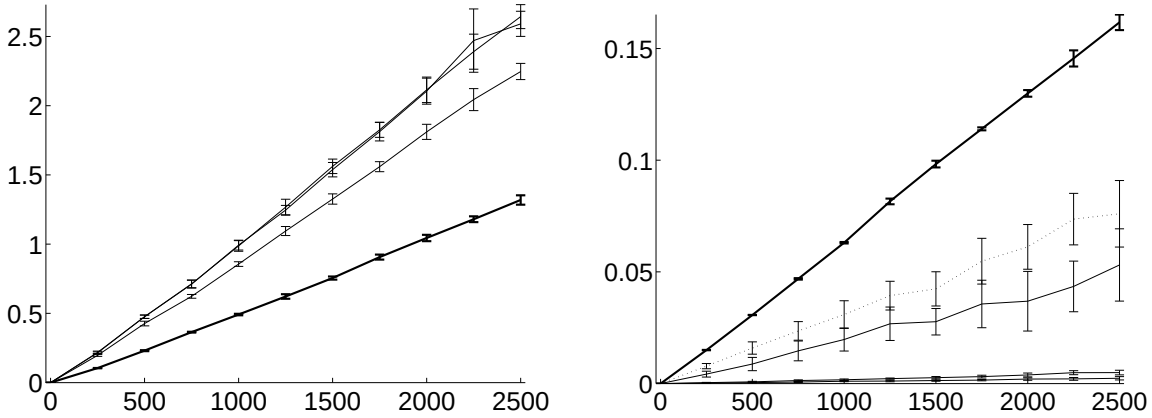

Figure 2.53: *Left:* Construction times (with standard deviations) for the data structure containing 2d line segments against their number for different end-point accuracies. The thick line corresponds to an end-point accuracy of $\sigma = 0.1$. The other lines correspond with increasing running times to the end-point accuracies $\sigma = 0.01, 0.001$ and 0.0001. *Right:* Query times (with standard deviations) for retrieving all incident 2d line segments against the number of 2d line segments contained in the data structure for different end-point accuracies. The thick line corresponds to a large end-point accuracy of $\sigma = 0.1$, the other solid lines with increasing performance to end-point accuracies of $\sigma = 0.01, 0.001$ and 0.0001. The dotted line shows the performance (with standard deviations) of sequential testing. Observe that the performance is worse than sequential testing for large uncertainties, because the algorithm has to descend in every leaf.

Chapter 3

Grouping

3.1 Motivation

As seen in section 1.2.4, grouping has been a key issue in computer vision from the very beginning. There are two major reasons for this. The first reason is the observation that the human visual system seems to group together visual stimuli in a very early stage of perception, enabling him to process the huge amount of input data. The second important reason is that many problems in vision can be formulated as grouping problems. After a precise notion of grouping is defined in section 3.2, some examples for this will be shown in section 3.3. For now an introductory example will be given informally, in order to ease the understanding of the formal definitions in the next section.

Given the image on the left hand side of figure 3.1, it is quite easy for a human observer to identify the depicted sketch of a house marked on the right hand side of figure 3.1 despite the presence of noise and outliers. Trying to replicate this human ability to identify groups in unordered observations is the goal of the grouping algorithms presented in the following.

While in the related field of perceptual grouping a number of criteria like simplicity, proximity, compactness, continuation or symmetry are identified as being responsible for this ability, the notion of grouping defined in the following will only be based on simplicity and proximity. The principle of compactness will also be exploited in a way by the entropy bound grouping approach presented in section 3.4.4.

Another important aspect for grouping is the use of prior knowledge. While the field of perceptual grouping, being sometimes considered as low-level image processing, is very reluctant to use this prior information, the notion of grouping presented in this work heavily depends on prior knowledge about the expected structure of the groups.

In the example presented above the prior knowledge would be that the sketch is made up of straight line segments. Furthermore simplicity demands that as few as possible line segments are present and at the same time most of the points are as close as possible to those line segments, which also serves the proximity requirement. In addition to that, the compactness requirement prevents the line segments to extend to far, as the points, which are not outliers, are expected to be compactly spaced along the segment.

In the subsequent sections the grouping problem will be defined formally followed by a range of exemplary grouping tasks in the presented framework. After this five different solution strategies will be presented and their performance will be compared on synthetic data. The application to the practical task of building reconstruction will be demonstrated chapter 4.

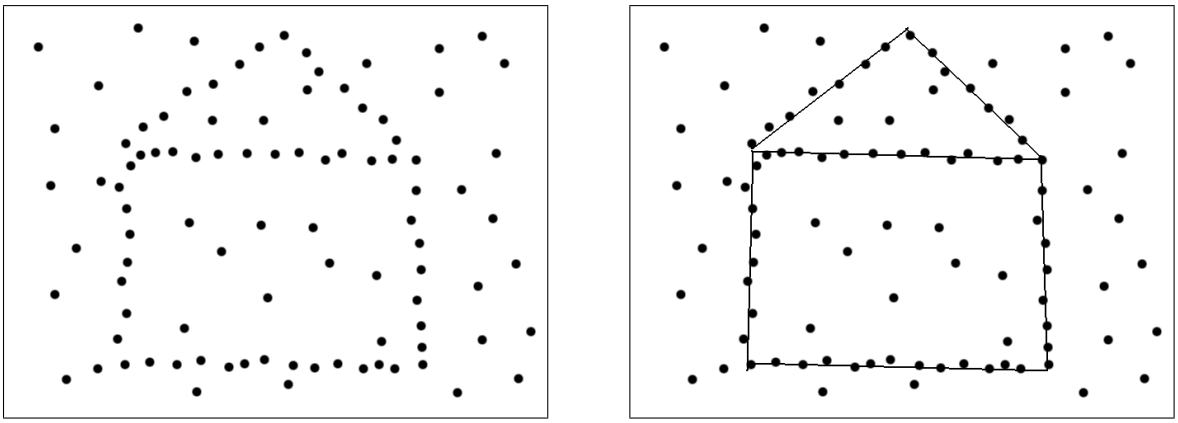

Figure 3.1: *Left:* An unordered set of 2d points. It is quite easy for a human observer, to identify a groups (along straight lines in this case) among those points. *Right:* The line segments associated with the groups yield a sketch of the depicted image.

3.2 Problem definition

In this section a generic problem definition for the grouping task will be given formally, which then will be made more specific in section 3.3. The definition is as follows: given is a finite set of observed, possibly heterogeneous entities

$$\mathcal{L} = \{(\boldsymbol{l}_1, \mathsf{C}_{l_1 l_1}), ..., (\boldsymbol{l}_N, \mathsf{C}_{l_N l_N})\} \tag{3.1}$$

being realizations of random variables drawn from a distribution having the unknown means

$$\tilde{\mathcal{L}} = \{\tilde{\boldsymbol{l}}_1, ..., \tilde{\boldsymbol{l}}_N\} \tag{3.2}$$

and known covariance matrices $\mathsf{C}_{l_1 l_1}, ..., \mathsf{C}_{l_N l_N}$. Note that this is exactly the representation of entities presented in section 2.1. In the example of the previous section this set of observations $\mathcal{L}$ contains the 2d points depicted in figure 3.1.

Next the prior knowledge about the internal structure of possible groups is modeled. Therefore a set of model functions

$$\{\boldsymbol{g}_1, ..., \boldsymbol{g}_N\} \tag{3.3}$$

describing the relation between the entities in terms of their relation with possible target models $\tilde{\boldsymbol{p}}$

$$g_i(\tilde{\boldsymbol{l}}_i, \tilde{\boldsymbol{p}}) = \mathbf{0} \tag{3.4}$$

is given. In the example of the previous section the target models $\tilde{\boldsymbol{p}}$ would be 2d lines and the group model functions $\boldsymbol{g}_i$ would be the scalar product of the entities represented as homogeneous vectors.

Now a grouping of the given entities can be defined as a set of group models

$$\tilde{\mathcal{P}} = \{\tilde{\boldsymbol{p}}_1, ..., \tilde{\boldsymbol{p}}_K\} \tag{3.5}$$

possibly required to fulfill a known internal model constraint

$$h(\tilde{\boldsymbol{p}}_i) = \mathbf{0} \tag{3.6}$$

and an assignment mapping of the entities onto those group models

$$\tilde{\zeta} : \{1, ..., N\} \rightarrow \{1, ...K\} \tag{3.7}$$

such that the model functions are fulfilled within each group

$$\tilde{\zeta}(i) = j \Rightarrow \boldsymbol{g}_i(\tilde{\boldsymbol{l}}_i, \tilde{\boldsymbol{p}}_j) = \boldsymbol{0} \tag{3.8}$$

In the example of the previous section, the set of group models $\tilde{\mathcal{P}}$ are the line segments depicted on the right hand side of figure 3.1. If the lines are represented as homogeneous vectors, then the internal model constraint $\boldsymbol{h}$ would be technically used to force their length to one. The assignment mapping $\tilde{\zeta}$ would be set for each point to point to its nearest line segment. Observe that the constraint is defined for the true entities $\tilde{\boldsymbol{l}}_i$ instead of the observed entities $\boldsymbol{l}_i$, so that the line must not pass through the points exactly.

This is a weak definition, as most problems allow multiple groupings, including the trivial and often non-unique solution of assigning a group for each entity, i.e. having $K = N$. It is therefore required to rank the groupings against each other and come up with some notion of good groupings. As a possible solution Occam's razor (cf. [Duda *et al.*, 2001, p.464]) suggests to favor simple explanations above complex ones, which means in the grouping context to prefer solutions with minimal number of groups K. Other criteria for identifying good groupings might be applicable, too. For instance it is often very important, to enforce some notion of compactness, i.e. require no single entity to lie far away from all other entities in the group. This latter criterion will be exploited in section 3.4.4.

The computation of such an optimal grouping poses two major challenges: first the complexity due to the combinatorics is exponential in the number of elements to be grouped and therefore prohibitive, and second the definition gives no hint on how to obtain the correct group models. Both of this tasks are coupled in the sense that knowing the correct group models usually makes it easy to assign the entities to the appropriate groups (e.g. using the methods for checking geometric relations presented in section 2.2) and on the other hand given the correct assignments it is usually possible to estimate group models using standard estimation techniques (cf. section 3.4.4, [Förstner and Wrobel, 2004, p.84] and [Heuel, 2001]). This property will be exploited by the expectation maximization approach in section 3.4.1.

In order to be able to devise grouping algorithms some technical assumptions on the unique computability of a group model must be made. Therefore one first defines a subset of entities

$$\tilde{l} \subset \tilde{\mathcal{L}} \tag{3.9}$$

as *consistent*, if there exists some group model $\tilde{\boldsymbol{p}}$, such that all of the constraints between the elements of $\tilde{l}$ and $\tilde{\boldsymbol{p}}$ are fulfilled.

In addition to this the subset $\tilde{l}$ is considered *sufficient*, if the entities contained in $\tilde{l}$ allow the unique computation of a group model $\tilde{\boldsymbol{p}}$ using the given constraints. In the geometric setting this means that an element $\tilde{\boldsymbol{p}}$ is constructible from the elements contained in the subset $\tilde{l}$, as has been demonstrated for many combinations of different element types in section 2.1.

Furthermore such a subset $\tilde{l}$ will be called *irreducible*, if the removal of any element will result in the subset not being sufficient any more. The set of all such minimal subsets will be denoted by

$$\tilde{\mathcal{S}} = \{\tilde{l} \subset \tilde{\mathcal{L}} | \tilde{l} \text{ is consistent, sufficient and irreducible}\} \tag{3.10}$$

By definition each element of this set implies a group model, with many of them being identical in the case of a small number of groups. In the absence of noise and outliers and

assuming that there exist no ambiguities in the solution, the set of all such implied group models directly yields the desired grouping. This property will be exploited by the graph based approaches in section 3.4.2.

As pointed out above, the true entities $\tilde{\mathcal{L}}$ are usually unknown, and one is given only observed entities denoted with

$$\mathcal{L} = \{(\boldsymbol{l}_1, \mathsf{C}_{l_1 l_1}), ..., (\boldsymbol{l}_N, \mathsf{C}_{l_N l_N})\} \tag{3.11}$$

being realizations of random variables drawn from a distribution having the unknown means

$$\tilde{\mathcal{L}} = \{\tilde{\boldsymbol{l}}_1, ..., \tilde{\boldsymbol{l}}_N\} \tag{3.12}$$

and known covariance matrices $\mathsf{C}_{l_1 l_1}, ..., \mathsf{C}_{l_N l_N}$. In this case the consistency requirement can be relaxed toward uncertain relations, so that also a set of consistent, sufficient and irreducible subsets of observations can be defined, which will be denoted as

$$\mathcal{S} = \{l \subset \mathcal{L} | l \text{ is consistent, sufficient and irreducible}\} \tag{3.13}$$

Of course the implied group models are in this case also uncertain entities and will be represented by the means and covariances of their distributions as usual.

Furthermore one must expect a possibly large number of outliers not following any known distribution to be present in the observations. Because the expectation maximization approach as well as the graph based approaches perform poorly under this conditions, an information theoretic approach, which balances robustness against efficiency, will be presented in section 3.4.4.

In the next section the notions defined abstractly in this section will be specialized toward specific geometric grouping problems that occur in computer vision.

3.3 Geometric grouping tasks

In the last section the problem of grouping was stated in a very generic framework. In the setting of uncertain oriented projective entities the abstract concepts can be made more specific, by requiring the group models $\tilde{\mathcal{P}}$ as well as the entities $\tilde{\mathcal{L}}$ all to be uncertain oriented projective entities as defined in section 2.1. The Boolean relations $R(\boldsymbol{l}, \boldsymbol{p}) \in \{0, 1\}$ between the entities presented in section 2.2 imply naive group model functions as

$$g(\boldsymbol{l}, \boldsymbol{p}) = 1 - R(\boldsymbol{l}, \boldsymbol{p}) \tag{3.14}$$

However, those naive group model functions take only values zero and one and are not very informative, as they do not encode any information about the distance between the two entities. Following the idea of Heuel [2001], in case of bi-linear relations (cf. equation (2.269))

$$R(\boldsymbol{l}, \boldsymbol{p}) = \begin{cases} 1 & \text{if } \mathsf{A}(\boldsymbol{l})\boldsymbol{p} = \mathsf{B}(\boldsymbol{p})\boldsymbol{l} = \boldsymbol{0} \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

a better choice as group model function is given by

$$\boldsymbol{g}(\boldsymbol{l}, \boldsymbol{p}) = \mathsf{A}(\boldsymbol{l})\boldsymbol{p} = \mathsf{B}(\boldsymbol{p})\boldsymbol{l} \tag{3.16}$$

where the Jacobians are immediately visible, which will turn out to be very useful later.

In the case of the bi-linear sign testing relations (cf. equation (2.272))

$$R(\boldsymbol{l},\boldsymbol{p}) = \begin{cases} 1 & \text{if } \boldsymbol{a}(\boldsymbol{l})\boldsymbol{p} = \boldsymbol{b}(\boldsymbol{p})\boldsymbol{l} \leq \boldsymbol{0} \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

the group model function would be

$$\boldsymbol{g}(\boldsymbol{l},\boldsymbol{p}) = \psi(\boldsymbol{a}(\boldsymbol{l})\boldsymbol{p}) = \psi(\boldsymbol{b}(\boldsymbol{p})\boldsymbol{l}) \tag{3.18}$$

with the function

$$\psi(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

This function is not bi-linear, but ignoring the discontinuity of ψ at zero, the Jacobians are also easily obtained as

$$\mathsf{A} = \frac{\partial \boldsymbol{g}(\boldsymbol{l},\boldsymbol{p})}{\partial \boldsymbol{p}} = \psi'(\boldsymbol{a}(\boldsymbol{l})\boldsymbol{p})\boldsymbol{a}(\boldsymbol{l}) = \begin{cases} \boldsymbol{a}(\boldsymbol{l}) & \text{if } \boldsymbol{a}(\boldsymbol{l})\boldsymbol{p} \geq 0 \\ \mathsf{0} & \text{otherwise} \end{cases} \tag{3.20}$$

and

$$\mathsf{B} = \frac{\partial \boldsymbol{g}(\boldsymbol{l},\boldsymbol{p})}{\partial \boldsymbol{l}} = \psi'(\boldsymbol{a}(\boldsymbol{l})\boldsymbol{p})\boldsymbol{b}(\boldsymbol{p}) = \begin{cases} \boldsymbol{b}(\boldsymbol{p}) & \text{if } \boldsymbol{a}(\boldsymbol{l})\boldsymbol{p} \geq 0 \\ \mathsf{0} & \text{otherwise} \end{cases} \tag{3.21}$$

with

$$\psi'(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.22}$$

Observe that $\mathsf{A} = \mathsf{B} = \mathsf{0}$, if the condition $R(\boldsymbol{l},\boldsymbol{p})$ is fulfilled at the point of linearization, so that the conditions can be discarded in this case.

If the relation is a conjunction of two relations $R_1(\boldsymbol{l},\boldsymbol{p})$ and $R_2(\boldsymbol{l},\boldsymbol{p})$

$$R(\boldsymbol{l},\boldsymbol{p}) = R_1(\boldsymbol{l},\boldsymbol{p}) \wedge R_2(\boldsymbol{l},\boldsymbol{p}) \tag{3.23}$$

corresponding to the group model functions $\boldsymbol{g}_1(\boldsymbol{l},\boldsymbol{p})$ and $\boldsymbol{g}_2(\boldsymbol{l},\boldsymbol{p})$, then the combined group model function would be obtained by stacking the two vectors as

$$\boldsymbol{g}(\boldsymbol{l},\boldsymbol{p}) = \begin{pmatrix} \boldsymbol{g}_1(\boldsymbol{l},\boldsymbol{p}) \\ \boldsymbol{g}_2(\boldsymbol{l},\boldsymbol{p}) \end{pmatrix} \tag{3.24}$$

Unfortunately the situation is not so easy for disjunctions. In case the relation is a disjunction of two relations $R_1(\boldsymbol{l},\boldsymbol{p})$ and $R_2(\boldsymbol{l},\boldsymbol{p})$

$$R(\boldsymbol{l},\boldsymbol{p}) = R_1(\boldsymbol{l},\boldsymbol{p}) \vee R_2(\boldsymbol{l},\boldsymbol{p}) \tag{3.25}$$

corresponding to the group model functions $\boldsymbol{g}_1(\boldsymbol{l},\boldsymbol{p})$ and $\boldsymbol{g}_2(\boldsymbol{l},\boldsymbol{p})$, then both of those group model functions have to be considered in subsequent processing. This will be denoted as

$$\boldsymbol{g}(\boldsymbol{l},\boldsymbol{p}) = \{\boldsymbol{g}_1(\boldsymbol{l},\boldsymbol{p}), \boldsymbol{g}_2(\boldsymbol{l},\boldsymbol{p})\} \tag{3.26}$$

In the following the components of the above grouping definition will be exemplified for a selection of useful grouping tasks occurring in computer vision. As many heterogeneous observations can be grouped into one group model, the following sections are ordered by group model type.

3.3.1 Grouping into 2d points

From incident 2d lines

Grouping elements into 2d points occurs in computer vision for instance when identifying vanishing points in images from uncertain oriented projective 2d line observations

$$\mathcal{L} = \{(\mathbf{l}_i, \mathsf{C}_{l_i l_i}), i = 1, ..., N\} \tag{3.27}$$

that intersect at this vanishing points. In this case the group model function is given by the bi-linear incidence relation between the 2d line and the 2d point (cf. equation (2.280))

$$g(\mathbf{l}_i, \mathbf{x}) = \mathbf{l}_i^T \mathbf{x} = \mathbf{x}^T \mathbf{l}_i \tag{3.28}$$

As a 2d point is uniquely constructible from the intersection of two 2d lines, every subset $l \subset \mathcal{L}$ having size $|l| \leq 2$ is consistent and every subset having size $|l| = 2$ is sufficient and irreducible. Hence, the set of consistent, sufficient and irreducible subsets of observations is given by

$$\mathcal{S} = \{\{\mathbf{l}_i, \mathbf{l}_j\} | i \neq j\} \tag{3.29}$$

where each element of this set defines as group model the intersection point of the two lines (cf. equation (2.38))

$$\mathbf{x}_{ij} = \mathsf{S}(\mathbf{l}_i)\mathbf{l}_j \tag{3.30}$$

having the covariance matrix

$$\mathsf{C}_{x_{ij}x_{ij}} = \mathsf{S}(\mathbf{l}_i)\mathsf{C}_{l_j l_j}\mathsf{S}(\mathbf{l}_i)^T + \mathsf{S}(\mathbf{l}_j)\mathsf{C}_{l_i l_i}\mathsf{S}(\mathbf{l}_j)^T \tag{3.31}$$

From incident 2d line segments

If the observed entities are uncertain oriented projective 2d line segments

$$\mathcal{L} = \{(\{\mathbf{l}_i, \mathbf{m}_i, \mathbf{n}_i\}, \{\mathsf{C}_{l_i l_i}, \mathsf{C}_{m_i m_i}, \mathsf{C}_{n_i n_i}\}), i = 1, ..., N\} \tag{3.32}$$

that have to intersect in some points, then the group model function is analogously given by the incidence relation between the 2d line segment and the 2d point (cf. equation (2.319))

$$\boldsymbol{g}(\{\mathbf{l}_i, \mathbf{m}_i, \mathbf{n}_i\}, \mathbf{x}) = \begin{pmatrix} \mathbf{l}_i^T \mathbf{x} \\ \psi(\mathbf{m}_i^T \mathbf{x}) \\ \psi(\mathbf{n}_i^T \mathbf{x}) \end{pmatrix} = \begin{pmatrix} \mathbf{x}^T \mathbf{l}_i \\ \psi(\mathbf{x}^T \mathbf{m}_i) \\ \psi(\mathbf{x}^T \mathbf{n}_i) \end{pmatrix} \tag{3.33}$$

Unlike the 2d line case, two 2d line segments not necessarily intersect. Every consistent subset $l \subset \mathcal{L}$ of size $|l| = 2$ is sufficient and irreducible for the same reason as before, though. Therefore the consistency requirement constraints the set of consistent, sufficient and irreducible subsets of observations to

$$\begin{aligned} \mathcal{S} = \{\{(\{\mathbf{l}_i, \mathbf{m}_i, \mathbf{n}_i\}), (\{\mathbf{l}_j, \mathbf{m}_j, \mathbf{n}_j\})\} \\ |i \neq j \wedge \mathsf{Intersect}(\{\mathbf{l}_i, \mathbf{m}_i, \mathbf{n}_i\}, \{\mathbf{l}_j, \mathbf{m}_j, \mathbf{n}_j\})\} \end{aligned} \tag{3.34}$$

in the case of 2d line segments. Again each element of this set induces as group model the intersection point

$$\mathbf{x}_{ij} = \mathsf{S}(\mathbf{l}_i)\mathbf{l}_j \tag{3.35}$$

having the covariance matrix

$$\mathsf{C}_{x_{ij}x_{ij}} = \mathcal{S}(\mathbf{l}_i)\mathsf{C}_{l_jl_j}\mathcal{S}(\mathbf{l}_i)^T + \mathcal{S}(\mathbf{l}_j)\mathsf{C}_{l_il_i}\mathcal{S}(\mathbf{l}_j)^T \tag{3.36}$$

Note that if both 2d lines and 2d line segments occur as observations, the set of consistent, sufficient and irreducible subsets can easily be augmented by the subsets containing all mixed pairs of intersecting lines and line segments.

From identical 2d points

It is always possible, to group equal entities together. In this case the set of observed entities contains the uncertain projective 2d points

$$\mathcal{L} = \{(\mathbf{x}_i, \mathsf{C}_{x_ix_i}), i = 1, ..., N\} \tag{3.37}$$

and the group model function is given by (cf. equation (2.292))

$$\boldsymbol{g}(\mathbf{x}_i, \mathbf{y}) = (\mathcal{S}(\mathbf{x}_i))^{[2]}\,\mathbf{y} = -\,(\mathcal{S}(\mathbf{y}))^{[2]}\,\mathbf{x}_i \tag{3.38}$$

The set of consistent, sufficient and irreducible subsets of observations is trivially given by

$$\mathcal{S} = \{\{\mathbf{x}_i\} | i = 1, ..., N\} \tag{3.39}$$

and the induced group model of each element of $\mathcal{S}$ is the point $\mathbf{x}_i$ itself.

3.3.2 Grouping into 2d lines

From incident 2d points

Grouping a set of uncertain oriented projective 2d points

$$\mathcal{L} = \{(\mathbf{x}_i, \mathsf{C}_{x_ix_i}), i = 1, ..., N\} \tag{3.40}$$

into straight lines is one of the key tasks encountered in feature extraction from images. It is also the toy example presented in section 3.1. Because it is the dual problem to grouping lines into points as discussed above, the model function for identifying co-linear points is given by (cf. equation (2.280))

$$g(\mathbf{x}_i, \mathbf{l}) = \mathbf{x}_i^T\mathbf{l} = \mathbf{l}^T\mathbf{x}_i \tag{3.41}$$

The set of consistent, sufficient and irreducible subsets is analogously given by

$$\mathcal{S} = \{\{\mathbf{x}_i, \mathbf{x}_j\} | i \neq j\} \tag{3.42}$$

where each element of this set defines as group model the connecting line between the two points (cf. equation (2.32))

$$\mathbf{l}_{ij} = \mathcal{S}(\mathbf{x}_i)\mathbf{x}_j \tag{3.43}$$

having the covariance matrix

$$\mathsf{C}_{l_{ij}l_{ij}} = \mathcal{S}(\mathbf{x}_i)\mathsf{C}_{x_jx_j}\mathcal{S}(\mathbf{x}_i)^T + \mathcal{S}(\mathbf{x}_j)\mathsf{C}_{x_ix_i}\mathcal{S}(\mathbf{x}_j)^T \tag{3.44}$$

From incident 2d edgels

Some feature extraction algorithms not only extract 2d point locations, but also gradient directions from the images. Hence, grouping uncertain oriented projective 2d edgels

$$\mathcal{L} = \{(\{\mathbf{x}_i, \mathbf{t}_i\}, \{\mathsf{C}_{x_i x_i}, \mathsf{C}_{t_i t_i}\}), i = 1, ..., N\} \tag{3.45}$$

without discarding the gradient information into 2d lines is an important step in feature extraction procedures, too. The group model function is here given by (cf. equation (2.333))

$$g(\{\mathbf{x}_i, \mathbf{t}_i\}, \mathbf{l}) = \begin{pmatrix} \mathbf{x}_i^T \\ \mathbf{t}_i^T C_\infty^T R_\perp^T \end{pmatrix} \mathbf{l} = \begin{pmatrix} \mathbf{l}^T & 0 \\ 0 & \mathbf{l}^T R_\perp C_\infty \end{pmatrix} \begin{pmatrix} \mathbf{x}_i \\ \mathbf{t}_i \end{pmatrix} \tag{3.46}$$

As a single edgel defines a unique 2d line (cf. equation (2.47)), the set of consistent, sufficient and irreducible subsets of observations is simply given by

$$\mathcal{S} = \{\{(\{\mathbf{x}_i, \mathbf{t}_i\})\} | i = 1, ..., N\} \tag{3.47}$$

with each such trivial subset implying a group model line according to equation (2.47).

From identical 2d lines

In case the set of observed entities contains the uncertain projective 2d lines

$$\mathcal{L} = \{(\mathbf{l}_i, \mathsf{C}_{l_i l_i}), i = 1, ..., N\} \tag{3.48}$$

that should be grouped into identical lines, the group model function is given by (cf. equation (2.293))

$$\boldsymbol{g}(\mathbf{l}_i, \mathbf{m}) = (\mathcal{S}(\mathbf{l}_i))^{[2]} \mathbf{m} = -(\mathcal{S}(\mathbf{m}))^{[2]} \mathbf{l}_i \tag{3.49}$$

The set of consistent, sufficient and irreducible subsets of observations is trivially given by

$$\mathcal{S} = \{\{\mathbf{l}_i\} | i = 1, ..., N\} \tag{3.50}$$

with the induced group model of each element being the line $\mathbf{l}_i$ itself.

3.3.3 Grouping into 3d points

From incident 3d lines

Grouping a set of uncertain oriented projective 3d lines

$$\mathcal{L} = \{(\mathbf{L}_i, \mathsf{C}_{L_i L_i}), i = 1, ..., N\} \tag{3.51}$$

into 3d points can be used for matching corresponding image points across multiple oriented views as has been presented in [Beder, 2004b] and will also be shown in section 4.2. The group model function is in this case given by (cf. equation (2.290))

$$\boldsymbol{g}(\mathbf{L}_i, \mathbf{X}) = \left(\overline{\mathbf{\Gamma}}(\mathbf{L}_i)^T)\right)^{[2]} \mathbf{X} = \left(\overline{\mathbf{\Pi}}(\mathbf{X})^T\right)^{[2]} \mathbf{L}_i \tag{3.52}$$

Because 3d lines do not need to intersect, a subset of lines can only be consistent, if all lines intersect in a single point. As two intersecting lines uniquely define a 3d point, the set of consistent, sufficient and irreducible subsets is given by (cf. equation (2.286))

$$\mathcal{S} = \{\{\mathbf{L}_i, \mathbf{L}_j\} | i \neq j \wedge \mathsf{Incident}(\mathbf{L}_i, \mathbf{L}_j)\} \tag{3.53}$$

Observe that in the feature matching application, this condition is equivalent to the epipolar constraint. The induced group models are the intersection points between the two lines, which have to be obtained using the estimation method proposed in [Heuel, 2001].

From incident 3d line segments

As pointed out in section 2.1.5, an image point feature may geometrically represent a 3d line segment, if a lower and an upper bound on the distance from the camera are known. The multi view matching problem can therefore be stated as a grouping problem of uncertain oriented projective 3d line segments

$$\mathcal{L} = \{(\{\mathbf{L}_i, \mathbf{E}_i, \mathbf{F}_i\}, \{\mathsf{C}_{L_iL_i}, \mathsf{C}_{E_iE_i}, \mathsf{C}_{F_iF_i}\}), i = 1, ..., N\} \tag{3.54}$$

intersecting at 3d points (cf. section 4.2). The group model function is then given by (cf. equation (2.356))

$$\boldsymbol{g}(\{\mathbf{L}_i, \mathbf{E}_i, \mathbf{F}_i\}, \mathbf{X}) = \begin{pmatrix} \left(\overline{\mathbf{\Gamma}}(\mathbf{L}_i)^T\right)^{[2]} \mathbf{X} \\ \psi(\mathbf{E}_i^T\mathbf{X}) \\ \psi(\mathbf{F}_i^T\mathbf{X}) \end{pmatrix} = \begin{pmatrix} \left(\overline{\mathbf{\Pi}}(\mathbf{X})^T\right)^{[2]} \mathbf{L}_i \\ \psi(\mathbf{X}^T\mathbf{E}_i) \\ \psi(\mathbf{X}^T\mathbf{F}_i^T) \end{pmatrix} \tag{3.55}$$

Applying the same reasoning as before in the case of 3d lines, the set of consistent, sufficient and irreducible subsets is given in this case by (cf. equation (2.365))

$$\begin{aligned} \mathcal{S} = \{\{(\{\mathbf{L}_i, \mathbf{E}_i, \mathbf{F}_i\}), (\{\mathbf{L}_j, \mathbf{E}_j, \mathbf{F}_j\})\} \\ |i \neq j \wedge \mathsf{Intersect}(\{\mathbf{L}_i, \mathbf{E}_i, \mathbf{F}_i\}, \{\mathbf{L}_j, \mathbf{E}_j, \mathbf{F}_j\})\} \end{aligned} \tag{3.56}$$

Observe that in the feature matching application, this constraint is equivalent to the epipolar constraint and the epipolar beam constraint (cf. [Heuel, 2004, p.161]). Again the induced group models are the intersection points between the two line segments and have to be obtained using the estimation method proposed in [Heuel, 2001].

From identical 3d points

In case a set of uncertain projective 3d points

$$\mathcal{L} = \{(\mathbf{X}_i, \mathsf{C}_{X_iX_i}), i = 1, ..., N\} \tag{3.57}$$

itself should be grouped, the group model function is given by (cf. equation (2.294))

$$\boldsymbol{g}(\mathbf{X}_i, \mathbf{Y}) = (\mathbf{\Pi}(\mathbf{X}_i))^{[3]} \mathbf{Y} = -(\mathbf{\Pi}(\mathbf{Y})))^{[3]} \mathbf{X}_i \tag{3.58}$$

The set of consistent, sufficient and irreducible subsets of observations is trivially given by

$$\mathcal{S} = \{\{\mathbf{X}_i\} | i = 1, ..., N\} \tag{3.59}$$

with the induced group model of each element being the point $\mathbf{X}_i$ itself.

3.3.4 Grouping into 3d lines

Grouping into 3d lines requires that the resulting entities are actual 3d lines and therefore fulfill the Plücker constraint (cf. equation (2.85)). This means that in this case a model constraint given by (cf. equation (2.85) and equation (2.93))

$$h(\mathbf{L}) = \mathbf{L}^T D_6 \mathbf{L} = 0 \tag{3.60}$$

has to be enforced.

From incident 3d points

As in the 2d case grouping uncertain oriented projective 3d points

$$\mathcal{L} = \{(\mathbf{X}_i, \mathsf{C}_{X_i X_i}), i = 1, ..., N\} \tag{3.61}$$

into straight lines is useful for identifying such linear structures in 3d point data. The group model function is given by (cf. eqaution (2.290))

$$g(\mathbf{X}_i, \mathbf{L}) = \left(\overline{\mathbf{\Pi}}(\mathbf{X}_i)^T\right)^{[2]} \mathbf{L} = \left(\overline{\mathbf{\Gamma}}(\mathbf{L})^T)\right)^{[2]} \mathbf{X}_i \tag{3.62}$$

As two 3d points uniquely define a 3d line like in the 2d case, the set of consistent, sufficient and irreducible subsets of observations if given by (cf. equation (2.83))

$$\mathcal{S} = \{\{\mathbf{X}_i, \mathbf{X}_j\} | i \neq j\} \tag{3.63}$$

where each element of this set defines as group model the connecting line between the two points (cf. equation (2.83))

$$\mathbf{L}_{ij} = \mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_j \tag{3.64}$$

having the covariance matrix

$$\mathsf{C}_{L_{ij}L_{ij}} = \mathbf{\Pi}(\mathbf{X}_i)\mathsf{C}_{X_j X_j}\mathbf{\Pi}(\mathbf{X}_i)^T + \mathbf{\Pi}(\mathbf{X}_j)\mathsf{C}_{X_i X_i}\mathbf{\Pi}(\mathbf{X}_j)^T \tag{3.65}$$

From incident 3d facets

If the 3d observations have an attached normal direction, the situation is more constrained and therefore more information is available for the grouping into straight lines. For grouping a set of uncertain oriented projective 3d facets

$$\mathcal{L} = \{(\{\mathbf{X}_i, \mathbf{T}_i\}, \{\mathsf{C}_{X_i X_i}, \mathsf{C}_{T_i T_i}\}), i = 1, ..., N\} \tag{3.66}$$

into straight 3d lines the group model function is given by (cf. eqaution (2.397))

$$\begin{aligned} \boldsymbol{g}(\{\mathbf{X}_i, \mathbf{T}_i\}, \mathbf{L}) &= \begin{pmatrix} \left(\overline{\mathbf{\Pi}}(\mathbf{X}_i)^T\right)^{[2]} \\ \mathbf{T}_i^T \begin{pmatrix} I_3 & 0_{3\times 3} \end{pmatrix} \end{pmatrix} \mathbf{L} & (3.67) \\ &= \begin{pmatrix} \left(\overline{\mathbf{\Gamma}}(\mathbf{L})^T)\right)^{[2]} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}^T \begin{pmatrix} I_3 \\ 0_{3\times 3} \end{pmatrix} \end{pmatrix} \begin{pmatrix} \mathbf{X}_i \\ \mathbf{T}_i \end{pmatrix} & (3.68) \end{aligned}$$

Due to the normals the situation is more constrained than the grouping from 3d points. A pair of 3d facets is only consistent, if it is co-linear. Therefore the set of consistent, sufficient and irreducible subsets of observations is given by (cf. equation (2.403))

$$\begin{aligned} \mathcal{S} = \{\{(\{\mathbf{X}_i, \mathbf{T}_i\}), (\{\mathbf{X}_j, \mathbf{T}_j\})\} & \\ |i \neq j \wedge \mathsf{Colinear}(\{\mathbf{X}_i, \mathbf{T}_i\}, \{\mathbf{X}_j, \mathbf{T}_j\})\} & \end{aligned} \tag{3.69}$$

The resulting group model is like the grouping from 3d points the connecting line between the two facets given by (cf. equation (2.83))

$$\mathbf{L}_{ij} = \mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_j \tag{3.70}$$

having the covariance matrix

$$\mathsf{C}_{L_{ij}L_{ij}} = \mathbf{\Pi}(\mathbf{X}_i)\mathsf{C}_{X_j X_j}\mathbf{\Pi}(\mathbf{X}_i)^T + \mathbf{\Pi}(\mathbf{X}_j)\mathsf{C}_{X_i X_i}\mathbf{\Pi}(\mathbf{X}_j)^T \tag{3.71}$$

From incident 3d planes

Grouping a set of uncertain oriented projective 3d planes

$$\mathcal{L} = \{(\mathbf{A}_i, \mathsf{C}_{A_i A_i}), i = 1, ..., N\} \tag{3.72}$$

into 3d lines can be used for matching image lines across multiple views as has been demonstrated in [Beder, 2004b] and will also be shown in section 4.2. The group model function is given by (cf. equation (2.291))

$$g(\mathbf{A}_i, \mathbf{L}) = \left(\mathbf{\Pi}(\mathbf{A}_i)^T\right)^{[2]} \mathbf{L} = \left(\mathbf{\Gamma}(\mathbf{L})^T\right)^{[2]} \mathbf{A}_i \tag{3.73}$$

Because this is dual to the problem of grouping points into straight lines and two planes uniquely define a 3d line, the set of consistent, sufficient and irreducible subsets of observations is given by

$$\mathcal{S} = \{\{\mathbf{A}_i, \mathbf{A}_j\} | i \neq j\} \tag{3.74}$$

where each element of this set defines as group model the line a the intersection of the two planes (cf. equation (2.109))

$$\mathbf{L}_{ij} = \overline{\mathbf{\Pi}}(\mathbf{A}_i)\mathbf{A}_j \tag{3.75}$$

having the covariance matrix

$$\mathsf{C}_{L_{ij}L_{ij}} = \overline{\mathbf{\Pi}}(\mathbf{A}_i)\mathsf{C}_{A_j A_j}\overline{\mathbf{\Pi}}(\mathbf{A}_i)^T + \overline{\mathbf{\Pi}}(\mathbf{A}_j)\mathsf{C}_{A_i A_i}\overline{\mathbf{\Pi}}(\mathbf{A}_j)^T \tag{3.76}$$

From incident convex 3d polygons

As pointed out in section 2.1.5, a line segment image feature may geometrically represent a 3d polygon, if a lower and an upper bound on the distance from the camera are known. The multi view matching problem of image line segments can therefore be stated as a grouping problem of a set of convex uncertain oriented projective 3d polygons

$$\begin{aligned} \mathcal{L} \;=\; & \{(\{\mathbf{X}_1^{(i)}, ..., \mathbf{X}_K^{(i)}\} \leftrightarrow \{\mathbf{A}^{(i)}, \mathbf{B}_1^{(i)}, ..., \mathbf{B}_K^{(i)}\}, \\ & \{\mathsf{C}_{X_1X_1}^{(i)}, ..., \mathsf{C}_{X_KX_K}^{(i)} \leftrightarrow \{\mathsf{C}_A^{(i)}, \mathsf{C}_{B_1B_1}^{(i)}, ..., \mathsf{C}_{B_KB_K}^{(i)}\}), i = 1, ..., N\} \end{aligned} \tag{3.77}$$

intersecting at 3d lines (cf. section 4.2), where the indices have been put up to improve readability. To derive the model function, equation (2.373) is converted to disjunctive normal form as follows (cf. equation (2.363), equation (2.361) and equation (2.362))

$$\begin{aligned} & \mathsf{Incident}(\mathbf{L}, \{\mathbf{X}_1, ..., \mathbf{X}_K\} \leftrightarrow \{\mathbf{A}, \mathbf{B}_1, ..., \mathbf{B}_K\}) && (3.78) \\ & \quad \Leftrightarrow \mathsf{Incident}(\mathbf{L}, \mathbf{A}) \wedge \bigvee_{j=1,...,K} \mathsf{Opposite}(\mathbf{X}_j, \mathbf{X}_{j+1}, \mathbf{L}) \\ & \quad \Leftrightarrow \bigvee_{j=1,...,K} \mathsf{Incident}(\mathbf{L}, \mathbf{A}) \wedge \mathsf{Opposite}(\mathbf{X}_j, \mathbf{X}_{j+1}, \overline{\boldsymbol{\gamma}}(\mathbf{L})) && (3.79) \\ & \quad \Leftrightarrow \bigvee_{j=1,...,K} \mathsf{Incident}(\mathbf{L}, \mathbf{A}) \wedge && (3.80) \\ & \qquad ((\mathsf{Left}(\mathbf{X}_j, \overline{\boldsymbol{\gamma}}(\mathbf{L})) \wedge \mathsf{Right}(\mathbf{X}_{j+1}, \overline{\boldsymbol{\gamma}}(\mathbf{L}))) \vee \\ & \qquad (\mathsf{Right}(\mathbf{X}_j, \overline{\boldsymbol{\gamma}}(\mathbf{L})) \wedge \mathsf{Left}(\mathbf{X}_{j+1}, \overline{\boldsymbol{\gamma}}(\mathbf{L})))) \\ & \quad \Leftrightarrow \bigvee_{j=1,...,K} \mathsf{Incident}(\mathbf{L}, \mathbf{A}) \wedge && (3.81) \end{aligned}$$

$$\begin{aligned} & ((\mathsf{Left}(\mathbf{X}_j, \overline{\boldsymbol{\gamma}}(\mathbf{L})) \wedge \mathsf{Right}(\mathbf{X}_{j+1}, \overline{\boldsymbol{\gamma}}(\mathbf{L}))) \\ \vee \bigvee_{j=1,...,K} & \mathsf{Incident}(\mathbf{L}, \mathbf{A}) \wedge \\ & ((\mathsf{Right}(\mathbf{X}_j, \overline{\boldsymbol{\gamma}}(\mathbf{L})) \wedge \mathsf{Left}(\mathbf{X}_{j+1}, \overline{\boldsymbol{\gamma}}(\mathbf{L}))) \end{aligned}$$

The set of possible alternative model functions is therefore given by (cf. equation (2.291))

$$\begin{aligned} & g(\{\mathbf{X}_1^{(i)}, ..., \mathbf{X}_K^{(i)}\} \leftrightarrow \{\mathbf{A}^{(i)}, \mathbf{B}_1^{(i)}, ..., \mathbf{B}_K^{(i)}\}, \mathbf{L}) \\ & = \left\{ \begin{pmatrix} \left(\boldsymbol{\Pi}(\mathbf{A}^{(i)})^T\right)^{[2]} \mathbf{L} \\ \psi(\pm \mathbf{X}_j^{(i)^T} \overline{\boldsymbol{\gamma}}(\mathbf{L})) \\ \psi(\mp \mathbf{X}_{j+1}^{(i)^T} \overline{\boldsymbol{\gamma}}(\mathbf{L})) \end{pmatrix} \right\} = \left\{ \begin{pmatrix} \left(\boldsymbol{\Gamma}(\mathbf{L})^T)\right)^{[2]} \mathbf{A}^{(i)} \\ \psi(\pm \overline{\boldsymbol{\gamma}}(\mathbf{L})^T \mathbf{X}_j^{(i)}) \\ \psi(\mp \overline{\boldsymbol{\gamma}}(\mathbf{L})^T \mathbf{X}_{j+1}^{(i)}) \end{pmatrix} \right\} \end{aligned} \tag{3.82}$$

Unlike planes not all 3d polygons actually intersect, so that the set of consistent, sufficient and irreducible subsets of observations is given by (cf. equation (2.382))

$$\begin{aligned} \mathcal{S} = \{\{(\{\mathbf{X}_1^{(i)}, ..., \mathbf{X}_K^{(i)}\}), (\{\mathbf{X}_1^{(j)}, ..., \mathbf{X}_K^{(j)}\})\} \\ |i \neq j \wedge \mathsf{Intersect}(\{\mathbf{X}_1^{(i)}, ..., \mathbf{X}_K^{(i)}\}, \{\mathbf{X}_1^{(j)}, ..., \mathbf{X}_K^{(j)}\})\} \end{aligned} \tag{3.83}$$

Observe that this constraint is equivalent to the epipolar beam constraint (cf. [Heuel, 2004, p.161]) in the image line segment matching application. The corresponding group models are obtained like in the case of grouping planes by intersecting the polygon planes from the plane representation of the two polygons (cf. equation (2.109))

$$\mathbf{L}_{ij} = \overline{\boldsymbol{\Pi}}(\mathbf{A}^{(i)}) \mathbf{A}^{(j)} \tag{3.84}$$

having the covariance matrix

$$\mathsf{C}_{L_{ij}L_{ij}} = \overline{\boldsymbol{\Pi}}(\mathbf{A}^{(i)}) \mathsf{C}_{A^{(j)}A^{(j)}} \overline{\boldsymbol{\Pi}}(\mathbf{A}^{(i)})^T + \overline{\boldsymbol{\Pi}}(\mathbf{A}^{(j)}) \mathsf{C}_{A^{(i)}A^{(i)}} \overline{\boldsymbol{\Pi}}(\mathbf{A}^{(j)})^T \tag{3.85}$$

From identical 3d lines

In case a set of uncertain projective 3d lines

$$\mathcal{L} = \{(\mathbf{L}_i, \mathsf{C}_{L_iL_i}), i = 1, ..., N\} \tag{3.86}$$

itself should be grouped, the group model function is given by (cf. equation (2.296))

$$g(\mathbf{L}_i, \mathbf{M}) = (\boldsymbol{\Delta}_j(\mathbf{L}_i))^{[4]} \mathbf{M} = -(\boldsymbol{\Delta}_j(\mathbf{M}))^{[4]} \mathbf{L}_i \tag{3.87}$$

The set of consistent, sufficient and irreducible subsets of observations is trivially given by

$$\mathcal{S} = \{\{\mathbf{L}_i\} | i = 1, ..., N\} \tag{3.88}$$

with the induced group model of each element being the line $\mathbf{L}_i$ itself.

3.3.5 Grouping into 3d planes

From incident 3d points

Grouping uncertain oriented projective 3d points

$$\mathcal{L} = \{(\mathbf{X}_i, \mathsf{C}_{X_i X_i}), i = 1, ..., N\} \tag{3.89}$$

into 3d planes is a common requirement, when for instance 3d point clouds obtained from a laser scanner are segmented. The group model function in this case is given by (cf. equation (2.283))

$$g(\mathbf{X}_i, \mathbf{A}) = \mathbf{X}_i^T \mathbf{A} = \mathbf{A}^T \mathbf{X}_i \tag{3.90}$$

For each subset of observations $l \subset \mathcal{L}$ containing at most three points, i.e. $|l| \leq 3$, there exists a plane going through all points. As three 3d points in general position define a 3d plane, the set of consistent, sufficient and irreducible subsets of observations is given by the set of all three element subsets of points

$$\mathcal{S} = \{\{\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k\} | i \neq j \neq k\} \tag{3.91}$$

The corresponding group models are the planes defined by the three points (cf. equation (2.102))

$$\mathbf{A}_{ijk} = \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_j)^T \mathbf{X}_k \tag{3.92}$$

having the covariance matrix

$$\begin{aligned} \mathsf{C}_{A_{ijk}A_{ijk}} &= \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_j)^T \mathsf{C}_{X_k X_k} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_i)\mathbf{X}_j \\ &\quad + \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_j)\mathbf{X}_k)^T \mathsf{C}_{X_i X_i} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_j)\mathbf{X}_k \\ &\quad + \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_k)\mathbf{X}_i)^T \mathsf{C}_{X_j X_j} \overline{\mathbf{\Gamma}}(\mathbf{\Pi}(\mathbf{X}_k)\mathbf{X}_i \end{aligned} \tag{3.93}$$

From incident 3d facets

In the laser scanning application often the normal directions are derived from the local neighborhood and 3d facets are given instead of raw 3d points. When grouping a set of uncertain oriented projective 3d facets

$$\mathcal{L} = \{(\{\mathbf{X}_i, \mathbf{T}_i\}, \{\mathsf{C}_{X_i X_i}, \mathsf{C}_{T_i T_i}\}), i = 1, ..., N\} \tag{3.94}$$

into 3d planes the group model function is given by (cf. equation (2.389))

$$\begin{aligned} g(\{\mathbf{X}_i, \mathbf{T}_i\}, \mathbf{A}) &= \begin{pmatrix} \mathbf{X}_i^T \\ \begin{pmatrix} \mathsf{S}(\mathbf{T}_i) & \mathbf{0} \end{pmatrix}^{[2]} \end{pmatrix} \mathbf{A} && (3.95) \\ &= \begin{pmatrix} \mathbf{A}^T & 0 \\ 0 & -(\mathsf{S}(\boldsymbol{A}_h))^{[2]} \end{pmatrix} \begin{pmatrix} \mathbf{X}_i \\ \mathbf{T}_i \end{pmatrix} && (3.96) \end{aligned}$$

As a single 3d facet defines a plane (cf. equation (2.118)) the set of consistent, sufficient and irreducible subsets of observations is trivially given by

$$\mathcal{S} = \{(\{\mathbf{X}_i, \mathbf{T}_i\}), i = 1, ..., N\} \tag{3.97}$$

and the corresponding group models are given by (cf. equation (2.118))

$$\mathbf{A} = \begin{pmatrix} X_h \boldsymbol{T} \\ -\boldsymbol{T}^T \boldsymbol{X}_0 \end{pmatrix} \tag{3.98}$$

with the covariance matrix being computed according to equation (2.121).

From incident 3d lines

If a set of uncertain oriented projective 3d lines

$$\mathcal{L} = \{(\mathbf{L}_i, \mathsf{C}_{L_i L_i}), i = 1, ..., N\} \tag{3.99}$$

has to be grouped into incident 3d planes, the group model function is given by (cf. equation (2.291))

$$g(\mathbf{L}_i, \mathbf{A}) = \left(\mathbf{\Gamma}(\mathbf{L}_i)^T)\right)^{[2]} \mathbf{A} = \left(\mathbf{\Pi}(\mathbf{A})^T\right)^{[2]} \mathbf{L}_i \tag{3.100}$$

This problem is dual to the grouping of 3d lines into 3d points and therefore consistent subsets have also to fulfill constraints. In this case two lines are only consistent, if they are co-planar, i.e. incident. Hence the set of consistent, sufficient and irreducible subsets of observations is given by (cf. equation (2.286))

$$\mathcal{S} = \{\{\mathbf{L}_i, \mathbf{L}_j\} | i \neq j \wedge \mathsf{Incident}(\mathbf{L}_i, \mathbf{L}_j)\} \tag{3.101}$$

The corresponding induced group models are the planes that are spanned by the pairs of co-planar lines and have to be obtained using the estimation method proposed in [Heuel, 2001] because of the additional constraint.

From identical 3d planes

In case a set of uncertain projective 3d planes

$$\mathcal{L} = \{(\mathbf{A}_i, \mathsf{C}_{A_i A_i}), i = 1, ..., N\} \tag{3.102}$$

itself should be grouped, the group model function is given by (cf. equation (2.295))

$$g(\mathbf{A}_i, \mathbf{B}) = \left(\overline{\mathbf{\Pi}}(\mathbf{A}_i)\right)^{[3]} \mathbf{B} = -\left(\overline{\mathbf{\Pi}}(\mathbf{B}))\right)^{[3]} \mathbf{A}_i \tag{3.103}$$

The set of consistent, sufficient and irreducible subsets of observations is trivially given by

$$\mathcal{S} = \{\{\mathbf{A}_i\} | i = 1, ..., N\} \tag{3.104}$$

with the induced group model of each element being the plane $\mathbf{A}_i$ itself.

Up to now a generic grouping framework has been presented and it has been shown, how various important problems in computer vision can be formulated in this framework. It has also been discussed briefly, how a notion of good groupings might be defined. However, no means of constructing such groupings in the first place is given so far. In the following several algorithms for solving this problem will be presented.

3.4 Grouping methods

In this section five different methods for finding groupings in the sense defined in section 3.2 will be presented. First, an expectation maximization scheme, which is only applicable in the case of a known number of groups will be presented. Then two graph based approaches, which work on the graph induced by the pairwise relations between the entities, will be developed. Next, a random sample consensus grouping method will be presented, that exploits the close connection between robust estimation and grouping. This connection is also used by the entropy bound approach presented thereafter, that constitutes one of the major contributions of this work.

3.4.1 Expectation maximization

In case the number of groups K is known in advance, the expectation maximization heuristic (cf. [Mitchell, 1997, p.191ff]) presents a very powerful technique for solving the grouping problem.

The basic idea goes as follows: starting from an initial, probably random, guess of group models together with a confidence represented as covariance matrix

$$\hat{\mathcal{P}}^{(0)} = \{(\hat{\boldsymbol{p}}_1^{(0)}, \mathsf{C}_{\hat{p}_1\hat{p}_1}^{(0)}), ..., (\hat{\boldsymbol{p}}_K^{(0)}, \mathsf{C}_{\hat{p}_K\hat{p}_K}^{(0)})\} \tag{3.105}$$

one computes the assignment function

$$\hat{\zeta}^{(\nu)}(i) = \text{argmin}_j \; d((\boldsymbol{l}_i, \mathsf{C}_{l_il_i}), (\boldsymbol{p}_j^{(\nu)}, \mathsf{C}_{p_jp_j}^{(\nu)})) \tag{3.106}$$

such that the distances of the observed entities and the current model are minimized. Using this assignment function, the group models $\hat{\boldsymbol{p}}_j^{(\nu+1)}$ are re-estimated from the set of observations, which has been assigned to this group

$$\mathcal{O}_j = \{\boldsymbol{l}_i | \zeta^{(\nu)}(i) = j\} \tag{3.107}$$

This process is iterated, until the group models and assignments do not change any more.

There are two issues to be addressed: first a suitable distance between the observations and the group models must be devised and second a method for estimating an optimal group model from the observations has to be given. As distance measure the Mahalanobis-distance has been used so far. Therefore the distance between a group model $\hat{\boldsymbol{p}}_j$ and an observation $\boldsymbol{l}_i$ will be measured using the Jacobians of the group model function

$$\mathsf{A}_{ij} = \left.\frac{\partial \boldsymbol{g}_i(\boldsymbol{l}_i, \boldsymbol{p}_j)}{\partial \boldsymbol{p}_j}\right|_{\hat{\boldsymbol{p}}_j^{(\nu)}, \boldsymbol{l}_i} \tag{3.108}$$

$$\mathsf{B}_{ij} = \left.\frac{\partial \boldsymbol{g}_i(\boldsymbol{l}_i, \boldsymbol{p}_j)}{\partial \boldsymbol{l}_i}\right|_{\hat{\boldsymbol{p}}_j^{(\nu)}, \boldsymbol{l}_i} \tag{3.109}$$

according to (cf. equation (2.270))

$$d(\boldsymbol{l}_i, \boldsymbol{p}_j) = \hat{\boldsymbol{p}}_j^T \mathsf{A}_{ij}^T \left(\mathsf{A}_{ij} \mathsf{C}_{l_il_i} \mathsf{A}_{ij}^T + \mathsf{B}_{ij} \mathsf{C}_{\hat{p}_j\hat{p}_j} \mathsf{B}_{ij}^T\right)^{-1} \mathsf{A}_{ij} \hat{\boldsymbol{p}}_j \tag{3.110}$$

Note that this measure is by definition consistent with the checking of relations presented in section 2.2, so that the data structure presented in section 2.3 can be used to improve the efficiency by considering for each group model only those observations that are consistent with the group model for some weak threshold.

The second issue is the estimation of a group model from a set of observations $\mathcal{O}_j$. This can be done using the group model functions by applying the standard Gauss-Helmert-model (cf. section 3.4.4 and [Förstner and Wrobel, 2004, p.84]), as has been proposed by [Heuel, 2001]. This estimation will be denoted with $\hat{\boldsymbol{p}}(\mathcal{O}_j)$ in the following.

3.4.2 Graph based approach

As pointed out before in section 3.2, each element in the set of consistent, sufficient and irreducible subsets of observations

$$\tilde{s} \in \tilde{\mathcal{S}} \tag{3.111}$$

defines a group model

$$\tilde{\boldsymbol{p}}(\tilde{s}) \tag{3.112}$$

by itself. In case of error-free observations, the set of group models is simply the union of those, i.e.

$$\tilde{\mathcal{P}} = \bigcup_{\tilde{s}\in\tilde{\mathcal{S}}} \{\tilde{\boldsymbol{p}}(\tilde{s})\} \tag{3.113}$$

In the presence of noise, however, the equality relation between the group models, which is implicit in the union operation, has to be made explicit. Therefore, one may construct the undirected graph

$$\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}) \tag{3.114}$$

with the vertex set being the set of consistent, sufficient and irreducible subsets

$$\tilde{\mathcal{V}} = \tilde{\mathcal{S}} \tag{3.115}$$

and the edge set being

$$\tilde{\mathcal{E}} = \{(\tilde{s}_1, \tilde{s}_2) \in \tilde{\mathcal{V}}^2 | \tilde{\boldsymbol{p}}(\tilde{s}_1) = \tilde{\boldsymbol{p}}(\tilde{s}_2)\} \tag{3.116}$$

which links two elements of $\tilde{\mathcal{S}}$ if and only if the corresponding group models are equal. The connected components of this graph are defined as the set

$$\begin{aligned} &\text{ConnectedComponents}(\tilde{\mathcal{G}}) = \\ &\quad \{V \subset \tilde{\mathcal{V}} | \forall v_1, v_2 \in V (\exists w_1 = v_1, ..., w_N = v_2 \in V (\\ &\qquad\qquad \forall 1 \leq i < N : (w_i, w_{i+1}) \in \tilde{\mathcal{E}})) \\ &\qquad \wedge \forall v \in \tilde{\mathcal{V}} \backslash V (\neg \exists w \in V ((v, w) \in \tilde{\mathcal{E}}))\} \end{aligned} \tag{3.117}$$

and a minimum clique partition of this graph is defined as

$$\begin{aligned} &\text{Cliques}(\tilde{\mathcal{G}}) \\ &= \text{argmin}_K \{V_i \subset \tilde{\mathcal{V}} | \dot{\bigcup}_{i=1}^{K} V_i = \tilde{\mathcal{V}} \wedge \forall v_1, v_2 \in V_i : \exists (v_1, v_2) \in \tilde{\mathcal{E}}\} \end{aligned} \tag{3.118}$$

For this particular graph it follows from the transitivity of the equality relation that those two sets are equal, i.e.

$$\text{ConnectedComponents}(\tilde{\mathcal{G}}) = \text{Cliques}(\tilde{\mathcal{G}}) \tag{3.119}$$

Furthermore by definition there is a one to one mapping between each such clique and a corresponding group model, i.e.

$$\tilde{\mathcal{P}} = \{\tilde{\boldsymbol{p}}(c_j) | c_j \in \{c_1, ..., c_K\} = \text{Cliques}(\tilde{\mathcal{G}})\} \tag{3.120}$$

and

$$\tilde{\zeta}(i) = j \Leftrightarrow \tilde{\boldsymbol{l}}_i \in c_j \tag{3.121}$$

so that finding the cliques, or equivalent connected components, in this noise-free graph is equivalent to finding the grouping.

In the presence of noise, the previous graph can be generalized as follows. The observed graph is now given by

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \tag{3.122}$$

with the vertex set being the set of consistent, sufficient and irreducible subsets of observations

$$\mathcal{V} = \mathcal{S} \tag{3.123}$$

and the edge set being

$$\mathcal{E} = \{(s_1, s_2) \in \mathcal{V}^2 | \mathsf{Equal}(\hat{\boldsymbol{p}}(s_1), \hat{\boldsymbol{p}}(s_2))\} \tag{3.124}$$

which links two elements of $\mathcal{S}$ if and only if the corresponding estimated group models fulfill the uncertain equality relation. Note that the construction of the graph is significantly sped up by the data structure presented in section 2.3. The connected components of this graph are now given by

$$\begin{aligned} &\text{ConnectedComponents}(\mathcal{G}) = \\ &\quad \{V \subset \mathcal{V} | \forall v_1, v_2 \in V (\exists w_1 = v_1, ..., w_N = v_2 \in V (\\ &\qquad \forall 1 \leq i < N : (w_i, w_{i+1}) \in \mathcal{E})) \\ &\qquad \wedge \forall v \in \mathcal{V} \backslash V (\neg \exists w \in V ((v, w) \in \mathcal{E}))\} \end{aligned} \tag{3.125}$$

and the minimum clique partition of this graph is

$$\begin{aligned} &\text{Cliques}(\mathcal{G}) \\ &= \text{argmin}_K \{V_i \subset \mathcal{V} | \dot{\bigcup_{i=1}^{K}} V_i = \mathcal{V} \wedge \forall v_1, v_2 \in V_i : \exists (v_1, v_2) \in \mathcal{E}\} \end{aligned} \tag{3.126}$$

Note that the uncertain equality relation is no longer transitive due to chaining effects, therefore in general the set of connected components is not equal to the set of cliques in general, i.e.

$$\text{ConnectedComponents}(\mathcal{G}) \neq \text{Cliques}(\mathcal{G}) \tag{3.127}$$

However, the set of connected components and the set of cliques both still directly correspond to group models. Hence, two grouping algorithms can be devised that in general come up with two different results. The first derives the set of group models from the set of connected components, i.e.

$$\hat{\mathcal{P}}_{CC} = \{\hat{\boldsymbol{p}}(cc_j) | cc_j \in \{cc_1, ..., cc_K\} = \text{ConnectedComponents}(\mathcal{G})\} \tag{3.128}$$

As a single observation can be contained in multiple consistent, sufficient and irreducible subsets of observations the corresponding assignment function is not unique. Therefore one either extends the model and allows a single observation to be assigned to multiple groups or one selects for each observation one of the set of groups it is contained in according to some criterion. As the size of the connected component indicates the strength of the group, this can be used as criterion yielding the following assignment function

$$\hat{\zeta}_{CC}(i) = \text{argmax}_j \{|cc_j| \ \mid \ \boldsymbol{l}_i \in cc_j\} \tag{3.129}$$

The second computes a minimum clique partition and derives the group models from this, i.e.

$$\hat{\mathcal{P}}_{Cl} = \{\hat{\boldsymbol{p}}(cl_j) | cl_j \in \{cl_1, ..., cl_K\} = \text{Cliques}(\mathcal{G})\} \tag{3.130}$$

with the corresponding assignment function being analogously defined as

$$\hat{\zeta}_{Cl}(i) = \text{argmax}_j \{|cl_j| \ \mid \ \boldsymbol{l}_i \in cl_j\} \tag{3.131}$$

The obvious disadvantage of the connected components solution is that one single outlier is able to link multiple groups together and therefore could severely distort the whole grouping result. Hence, one has to take extra care in order to avoid this or go for the clique partition approach. As all vertices within a group have to have a linking edge in this approach, it could be expected to react more robustly toward outliers than the connected components approach, as groups containing such an outlier are more likely to be split up. In [Beder, 2004b] the clique partition approach has been presented for the task of matching points and lines across multiple oriented images.

While the connected components of a graph are computable in polynomial time, the mimimum clique partition problem is unfortunately known to be NP-hard. Therefore one has to come up with an approximation algorithm. A successful heuristic in the point an line matching application has been to sort the vertices by their degree and then greedily accumulate vertices into cliques. This heuristic tends to produce large cliques in the beginning, which are more likely to be inliers and then produces smaller cliques, which usually correspond to the outliers that have not much support by other observations. However, an outlier vertex linking two cliques together also has a high degree, so that this heuristic is by no means fool-proof and has to be augmented by other heuristics, such as eliminating vertices of high uncertainty in the first place that are more likely to behave like this.

3.4.3 Random Sample Consensus method

The applicability of the graph based grouping methods described in the previous section suffers from the computational complexity. This complexity is not only incurred by the NP-hardness of the clique partition problem, but also by the problem size itself. The number of consistent, sufficient and irreducible subsets of observations, hence the size of the problem instance itself, is potentially exponential in the number of oberservations involved. In addition to this the number of edges of the graph can be quadratic in the number of those subsets. While the latter issue has been partly addressed by the data structure presented in section 2.3, the large number of consistent, sufficient and irreducible subsets can affect the practical applicability of the presented method.

In order to circumvent this problem, not all consistent, sufficient and irreducible subsets have to be computed. Instead they can be sampled at random and a consensus set of observations that fit the corresponding group model can be computed. After a certain amount of iterations the largest consensus set is selected, all observations are removed and the process is iterated. This method has been used in [Beder and Förstner, 2006a] and [Beder and Förstner, 2006b] for grouping 3d points into cylinders.

Although this procedure has successfully been applied to robust estimation problems (cf. [Fischler and Bolles, 1981]) it suffers two major disadvantages in the grouping context. First, the required number of iterations heavily depends on the expected number of groups, so that there is no improvement on the running time over the graph based methods presented in the previous section, as nearly the whole set of consistent, sufficient and irreducible subsets has to be sampled, in order to obtain reasonable results. The second severe disadvantage of this method is that the whole consensus set is considered at once. In the presence of geometrically weak consistent, sufficient and irreducible subsets of observations or leverage points inside the consensus set, the procedure breaks down. A solution to this problem would be to carefully and incrementally build the consensus set. This idea will be exploited in the following section.

3.4.4 Entropy bound approach

The grouping approaches presented so far all have some caveat: The expectation maximization heuristic requires the number of groups as well as some initial guess of group models to be known in advance, which is impractical in most applications. The connected component grouping algorithm is extremely unstable in the presence of outliers, while the clique partition grouping algorithm is NP-hard and therefore requires exponential running time.

In the following an alternative grouping algorithm will be derived. Incremental estimation is analyzed in order to devise an information theoretic measure that allows to decide for one single observation, if it should be grouped together with an existing group. Using this measure enables a greedy grouping strategy that is able to balance robustness against efficiency.

In order to devise this novel grouping algorithm, first the effect of adding a group of observations into an estimation will be derived. Afterward the entropy change incurred by this new group of observations will be analyzed and it will be shown to comprise of two additive parts, which reflect the two aspects of similarity and proximity of the new observations with respect to the old observations. Finally a grouping algorithm will be devised, that uses those two measures in order to guide the grouping process in a robust and efficient manner.

Incremental Gauss-Helmert-model with restrictions

When adding a new set of observations to an existing set of observations, the resulting group model changes. In order to analyze this change, the effect of adding a second group of observations on the estimation of a group model in the Gauss-Helmert-model will be studied in the following.

The general Gauss-Helmert-model (cf. [Mikhail and Ackermann, 1976, p.214] and [Förstner and Wrobel, 2004, p.84]) for two sets of independent observations is given by the constraints

$$\boldsymbol{g}_1(\tilde{\boldsymbol{l}_1}, \tilde{\boldsymbol{p}}) = \boldsymbol{0} \tag{3.132}$$

$$\boldsymbol{g}_2(\tilde{\boldsymbol{l}_2}, \tilde{\boldsymbol{p}}) = \boldsymbol{0} \tag{3.133}$$

between the true parameters $\tilde{\boldsymbol{p}}$ and the true, i.e. error-free, observations $\tilde{\boldsymbol{l}_1}$ and $\tilde{\boldsymbol{l}_2}$ together with the gauge-constraints

$$\boldsymbol{h}(\tilde{\boldsymbol{p}}) = \boldsymbol{0} \tag{3.134}$$

on the parameters. In the following the number of parameters is U, the number of observations in the first set is N_1, the number of observations in the second set is N_2 and the number of gauge-constraints is H. Furthermore the actual observations $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ are assumed to be realizations of random variables drawn from distributions with means $\tilde{\boldsymbol{l}_1}$ and $\tilde{\boldsymbol{l}_2}$ and covariances ${\sigma_0}^2 \mathsf{C}_{11}$ and ${\sigma_0}^2 \mathsf{C}_{22}$ of known shape and unknown magnitude. The best linear unbiased estimate $\hat{\boldsymbol{p}}_1$ of the parameters from the first set of observations $\boldsymbol{l}_1$ together with its covariance matrix $\hat{\mathsf{C}}_{\hat{p}_1\hat{p}_1}$ is computable as follows. First, using the initial values $\hat{\boldsymbol{p}}_1^{(\nu)} = \hat{\boldsymbol{p}}_1^{DLT}$ and $\hat{\boldsymbol{l}}_1^{(\nu)} = \boldsymbol{l}_1$, the Jacobians

$$\mathsf{A}_1 = \left.\frac{\partial \boldsymbol{g}_1(\boldsymbol{l}_1, \boldsymbol{p})}{\partial \boldsymbol{p}}\right|_{\hat{\boldsymbol{p}}_1^{(\nu)}, \hat{\boldsymbol{l}}_1^{(\nu)}} \tag{3.135}$$

$$\mathsf{B}_1 = \left.\frac{\partial \boldsymbol{g}_1(\boldsymbol{l}_1, \boldsymbol{p})}{\partial \boldsymbol{l}_1}\right|_{\hat{\boldsymbol{p}}_1^{(\nu)}, \hat{\boldsymbol{l}}_1^{(\nu)}} \tag{3.136}$$

$$\mathsf{H} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{p})}{\partial \boldsymbol{p}}\right|_{\hat{\boldsymbol{p}}_1^{(\nu)}} \tag{3.137}$$

and the contradictions

$$\boldsymbol{c}_{g_1} = -\boldsymbol{g}_1(\hat{\boldsymbol{l}}_1^{(\nu)}, \hat{\boldsymbol{p}}_1^{(\nu)}) - B_1(\boldsymbol{l}_1 - \hat{\boldsymbol{l}}_1^{(\nu)}) \tag{3.138}$$

$$\boldsymbol{c}_h = -\boldsymbol{h}(\hat{\boldsymbol{p}}_1^{(\nu)}) \tag{3.139}$$

are computed. Using the abbreviations

$$W_{11} = B_1 C_{11} B_1^T \tag{3.140}$$

$$\boldsymbol{y}_1 = A_1^T W_{11}^{-1} \boldsymbol{c}_{g_1} \tag{3.141}$$

the normal equation system (cf. [Förstner and Wrobel, 2004, p.85]) for the first set of observations is

$$\begin{pmatrix} A_1^T W_{11}^{-1} A_1 & H^T \\ H & 0 \end{pmatrix} \begin{pmatrix} \Delta\hat{\boldsymbol{p}}_1 \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{c}_h \end{pmatrix} \tag{3.142}$$

Under the assumption that the rows of H are orthonormal and prependicular to the rows of the normal equation matrix $H \perp A_1^T W_{11}^{-1} A_1$, the solution (cf. [Niemeier, 2001, p.245]) is given by

$$\begin{pmatrix} \Delta\hat{\boldsymbol{p}}_1 \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} A_1^T W_{11}^{-1} A_1 & H^T \\ H & 0 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{c}_h \end{pmatrix} \tag{3.143}$$

$$= \begin{pmatrix} (A_1^T W_{11}^{-1} A_1 + H^T H)^{-1} - H^T H & H^T \\ H & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{c}_h \end{pmatrix} \tag{3.144}$$

In case of a non-linear gauge-constraint $\boldsymbol{h}$, it is numerically more stable, to enforce the above requirement by projecting the design matrix according to

$$A_1' = A_1(I - H^T H) \tag{3.145}$$

Now the expected covariance matrix is

$$C_{\hat{p}_1\hat{p}_1} = (A_1^T W_{11}^{-1} A_1 + H^T H)^{-1} - H^T H \tag{3.146}$$

and the estimated parameters are

$$\hat{\boldsymbol{p}}_1^{(\nu+1)} = \hat{\boldsymbol{p}}_1^{(\nu)} + \Delta\hat{\boldsymbol{p}}_1 \tag{3.147}$$

with

$$\Delta\hat{\boldsymbol{p}}_1 = C_{\hat{p}_1\hat{p}_1} \boldsymbol{y}_1 + H^T \boldsymbol{c}_h \tag{3.148}$$

The residuals (cf. [Förstner and Wrobel, 2004, p.85]) are given by

$$\hat{\boldsymbol{v}}_1 = -C_{11} B_1^T W_{11}^{-1} (A_1 \Delta\hat{\boldsymbol{p}}_1 - \boldsymbol{c}_{g_1}) \tag{3.149}$$

and therefore the adjusted observations are

$$\hat{\boldsymbol{l}}_1^{(\nu+1)} = \boldsymbol{l}_1 + \hat{\boldsymbol{v}}_1 \tag{3.150}$$

If the observations must fulfill some constraints, they can be imposed on the adjusted observations at this point (cf. [Heuel, 2004, p.193]). In case of non-linear constraints, this process

must be iterated using $\hat{\boldsymbol{p}}_1^{(\nu+1)}$ and $\hat{\boldsymbol{l}}_1^{(\nu+1)}$ a new initial values. Finally the weighted sum of the squared residuals (cf. [Heuel, 2004, p.193]) is obtained as

$$\Omega_1^2 = \hat{\boldsymbol{v}}_1^T B_1^T W_{11}^{-1} B_1 \hat{\boldsymbol{v}}_1 \tag{3.151}$$

and thus the estimated covariance matrix from the first set of parameters is

$$\hat{C}_{\hat{p}_1\hat{p}_1} = \hat{\sigma_1}^2 C_{\hat{p}_1\hat{p}_1} \tag{3.152}$$

with

$$\hat{\sigma}_1^2 = \frac{\Omega_1^2}{N_1 - U + H} \tag{3.153}$$

Now the second set of observations should be included into the estimation. Therefore using the initial values $\hat{\boldsymbol{p}}_2^{(\nu)} = \hat{\boldsymbol{p}}_1^{(\nu)}$ and $\hat{\boldsymbol{l}}_2^{(\nu)} = \boldsymbol{l}_2$, the Jacobians

$$A_2 = \left.\frac{\partial \boldsymbol{g}_2(\boldsymbol{l}_2, \boldsymbol{p})}{\partial \boldsymbol{p}}\right|_{\hat{\boldsymbol{p}}_2^{(\nu)}, \hat{\boldsymbol{l}}_2^{(\nu)}} \tag{3.154}$$

$$B_2 = \left.\frac{\partial \boldsymbol{g}_2(\boldsymbol{l}_2, \boldsymbol{p})}{\partial \boldsymbol{l}_2}\right|_{\hat{\boldsymbol{p}}_2^{(\nu)}, \hat{\boldsymbol{l}}_2^{(\nu)}} \tag{3.155}$$

$$H = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{p})}{\partial \boldsymbol{p}}\right|_{\hat{\boldsymbol{p}}_2^{(\nu)}} \tag{3.156}$$

and the contradictions

$$\boldsymbol{c}_{g_2} = -\boldsymbol{g}_2(\hat{\boldsymbol{l}}_2^{(\nu)}, \hat{\boldsymbol{p}}_2^{(\nu)}) - B_2(\boldsymbol{l}_2 - \hat{\boldsymbol{l}}_2^{(\nu)}) \tag{3.157}$$

$$\boldsymbol{c}_h = -\boldsymbol{h}(\hat{\boldsymbol{p}}_2^{(\nu)}) \tag{3.158}$$

are computed. Under the assumption that the constraint equations for the two observation sets are independent of the observations of the other set, i.e.

$$\frac{\partial \boldsymbol{g}_1(\boldsymbol{l}_1, \boldsymbol{p})}{\partial \boldsymbol{l}_2} = 0 \tag{3.159}$$

and

$$\frac{\partial \boldsymbol{g}_2(\boldsymbol{l}_2, \boldsymbol{p})}{\partial \boldsymbol{l}_1} = 0 \tag{3.160}$$

and that the two observation sets are not correlated, i.e. $C_{12} = C_{21} = 0$, the normal equation matrix for the joint estimation from both sets of observations can be partitioned as (cf. [Steffen, 2006])

$$\begin{pmatrix} A_1^T & A_2^T \end{pmatrix} \left(\begin{pmatrix} B_1^T & 0 \\ 0 & B_2^T \end{pmatrix} \begin{pmatrix} C_{11} & 0 \\ 0 & C_{22} \end{pmatrix} \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix} \right)^{-1} \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \tag{3.161}$$
$$= A_1^T (B_1^T C_{11} B_1)^{-1} A_1 + A_2^T (B_2^T C_{22} B_2)^{-1} A_2$$

Further observe that the contradictions for the first set of observations change due to the new parameter vector, so that

$$\bar{\boldsymbol{c}}_{g_1} = -\boldsymbol{g}_1(\hat{\boldsymbol{l}}_1, \hat{\boldsymbol{p}}_2^{(\nu)}) - B_1(\boldsymbol{l}_1 - \hat{\boldsymbol{l}}_1) \tag{3.162}$$

$$= -\boldsymbol{g}_1(\hat{\boldsymbol{l}}_1, \hat{\boldsymbol{p}}_2^{(\nu)}) + B_1 \hat{\boldsymbol{v}}_1 \tag{3.163}$$

and thus

$$\begin{aligned}
\bar{\boldsymbol{y}}_1 &= A_1^T W_{11}^{-1} \bar{\boldsymbol{c}}_{g_1} && (3.164)\\
&= -A_1^T W_{11}^{-1} \boldsymbol{g}_1(\hat{\boldsymbol{l}}_1^{(\nu)}, \hat{\boldsymbol{p}}_2^{(\nu)}) + A_1 W_{11}^{-1} B_1 \hat{\boldsymbol{v}}_1 && (3.165)\\
&= -A_1^T W_{11}^{-1} \boldsymbol{g}_1(\hat{\boldsymbol{l}}_1^{(\nu)}, \hat{\boldsymbol{p}}_2^{(\nu)}) && (3.166)\\
&\quad -A_1 W_{11}^{-1} B_1 C_{11} B_1^T W_{11}^{-1}(A_1 \Delta\hat{\boldsymbol{p}}_1 - \boldsymbol{c}_{g_1}) \\
&= -A_1^T W_{11}^{-1} \boldsymbol{g}_1(\hat{\boldsymbol{l}}_1^{(\nu)}, \hat{\boldsymbol{p}}_2^{(\nu)}) - A_1 W_{11}^{-1}(A_1 \Delta\hat{\boldsymbol{p}}_1 - \boldsymbol{c}_{g_1}) && (3.167)\\
&= -(A_1^T W_{11}^{-1} \boldsymbol{g}_1(\hat{\boldsymbol{l}}_1^{(\nu)}, \hat{\boldsymbol{p}}_2^{(\nu)}) - \underbrace{(A_1 W_{11}^{-1} A_1 \Delta\hat{\boldsymbol{p}}_1 - \boldsymbol{y}_1)}_{=\boldsymbol{0}} && (3.168)\\
&= -A_1^T W_{11}^{-1} \boldsymbol{g}_1(\hat{\boldsymbol{l}}_1^{(\nu)}, \hat{\boldsymbol{p}}_2^{(\nu)}) && (3.169)
\end{aligned}$$

where the last equality follows from the normal equation system. Note that the change of residuals for the first set of observations, like for the estimation of the new variance factor (see below), is not taken into account. Hence, using the abreviations

$$W_{22} = B_2 C_{22} B_2^T \tag{3.170}$$

$$\boldsymbol{y}_2 = \bar{\boldsymbol{y}}_1 + A_2^T W_{22}^{-1} \boldsymbol{c}_{g_2} \tag{3.171}$$

the new normal equation system becomes

$$\begin{pmatrix} A_1^T W_{11}^{-1} A_1 + A_2^T W_{22}^{-1} A_2 & H^T \\ H & 0 \end{pmatrix} \begin{pmatrix} \Delta\hat{\boldsymbol{p}}_2 \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}_2 \\ \boldsymbol{c}_h \end{pmatrix} \tag{3.172}$$

Using again the orthonormality of the rows of H and the perpendicularity of $H \perp (A_1^T W_{11}^{-1} A_1 + A_2^T W_{22}^{-1} A_2)$, the solution is

$$\begin{aligned}
\begin{pmatrix} \Delta\hat{\boldsymbol{p}}_2 \\ \boldsymbol{\mu} \end{pmatrix} &= \begin{pmatrix} A_1^T W_{11}^{-1} A_1 + A_2^T W_{22}^{-1} A_2 & H^T \\ H & 0 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{y}_2 \\ \boldsymbol{c}_h \end{pmatrix} && (3.173)\\
&= \begin{pmatrix} (A_1^T W_{11}^{-1} A_1 + A_2^T W_{22}^{-1} A_2 + H^T H)^{-1} - H^T H & H^T \\ H & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{y}_2 \\ \boldsymbol{c}_h \end{pmatrix} && (3.174)
\end{aligned}$$

Again in case of a non-linear gauge-constraint $\boldsymbol{h}$, it is numerically more stable, to enforce the above requirement by projecting the design matrices and covaraiance matrix according to

$$A_1' = A_1(I - H^T H) \tag{3.175}$$

$$A_2' = A_2(I - H^T H) \tag{3.176}$$

$$C'_{\hat{p}_1\hat{p}_1} = (I - H^T H) C_{\hat{p}_1\hat{p}_1} (I - H^T H) \tag{3.177}$$

Now using the well-known matrix identity (cf. [Koch, 1997, p.37])

$$(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} \tag{3.178}$$

the expected covariance matrix is

$$\begin{aligned}
C_{\hat{p}_2\hat{p}_2} &= (A_1^T W_{11}^{-1} A_1 + H^T H + A_2^T W_{22}^{-1} A_2)^{-1} - H^T H && (3.179)\\
&= (A_1^T W_{11}^{-1} A_1 + H^T H)^{-1} && (3.180)
\end{aligned}$$

$$
\begin{aligned}
& -(A_1^T W_{11}^{-1} A_1 + H^T H)^{-1} \\
& \quad A_2^T (W_{22} + A_2 (A_1^T W_{11}^{-1} A_1 + H^T H)^{-1} A_2^T)^{-1} A_2 \\
& \quad (A_1^T W_{11}^{-1} A_1 + H^T H)^{-1} - H^T H \\
= \; & (C_{\hat{p}_1 \hat{p}_1} + H^T H) - H^T H \\
& -(C_{\hat{p}_1 \hat{p}_1} + H^T H) A_2^T (W_{22} + A_2 (C_{\hat{p}_1 \hat{p}_1} + H^T H) A_2^T)^{-1} \\
& \quad A_2 (C_{\hat{p}_1 \hat{p}_1} + H^T H)
\end{aligned} \tag{3.181}
$$

where the last equality is due to equation (3.146). Putting everything together, the new covariance matrix of the parameters estimated from both sets of observations is

$$C_{\hat{p}_2 \hat{p}_2} = C_{\hat{p}_1 \hat{p}_1} - F A_2 (C_{\hat{p}_1 \hat{p}_1} + H^T H) \tag{3.182}$$

with

$$F = (C_{\hat{p}_1 \hat{p}_1} + H^T H) A_2^T (W_{22} + A_2 (C_{\hat{p}_1 \hat{p}_1} + H^T H) A_2^T)^{-1} \tag{3.183}$$

The estimated parameters itself are obtained as

$$\hat{\boldsymbol{p}}_2^{(\nu+1)} = \hat{\boldsymbol{p}}_2^{(\nu)} + \Delta \hat{\boldsymbol{p}}_2 \tag{3.184}$$

with

$$\Delta \hat{\boldsymbol{p}}_2 = C_{\hat{p}_2 \hat{p}_2} \boldsymbol{y}_2 + H^T \boldsymbol{c}_h \tag{3.185}$$

and the residuals of the joint estimation for the second set of observations are

$$\hat{\boldsymbol{v}}_2 = -C_{22} B_2^T W_{22}^{-1} (A_2 \Delta \hat{\boldsymbol{p}}_2 - \boldsymbol{c}_{g_2}) \tag{3.186}$$

so that the adjusted observations are

$$\hat{\boldsymbol{l}}_2^{(\nu+1)} = \boldsymbol{l}_2 + \hat{\boldsymbol{v}}_2 \tag{3.187}$$

Again, in case of non-linear constraints, this process must be iterated using $\hat{\boldsymbol{p}}_2^{(\nu+1)}$ and $\hat{\boldsymbol{l}}_2^{(\nu+1)}$ as new initial values. Note that in this case the solution is only an approximation, since the linearization is made at a different parameter vector, so that the Jacobians of the first estimation might no longer be accurate, if the cost function is too rough in the range of the two estimated parameters. Finally, following the idea of Förstner and Wrobel [2004, p.93] and neglecting the change of residuals for the first set of observations, the weighted sum of the squared residuals may be updated as

$$\Omega_2^2 = \Omega_1^2 + \hat{\boldsymbol{v}}_2^T B_2^T W_{22}^{-1} B_2 \hat{\boldsymbol{v}}_2 \tag{3.188}$$

so that the estimated covariance of the parameters from both sets of observations is

$$\hat{C}_{\hat{p}_2 \hat{p}_2} = \hat{\sigma}_2^2 C_{\hat{p}_2 \hat{p}_2} \tag{3.189}$$

with

$$\hat{\sigma}_2^2 = \frac{\Omega_2^2}{N_1 + N_2 - U + H} \tag{3.190}$$

Entropy change incurred by the incremental estimation

In the previous section the effect of adding a new set of observations on the mean and covariance of the distribution of the resulting group model has been analyzed. In the following the effect on the entropy of this group model distribution will be studied. The minimum-description-length principle demands that the overall entropy shall be minimized, so that grouping together observations, which promise an entropy decrease, should be favored. On the other hand rapid entropy decrease endangers greedy algorithms to make fatal decisions that cannot be reversed. As a solution, the analysis of the change of entropy gives some useful insights, how this trade-off between efficiency and robustness can be controlled.

Let the k-dimensional subspace determinant of a singular matrix C with $\mathrm{rank}\mathsf{C} = k$ be defined as the product of its non-zero eigenvalues

$$|\mathsf{C}|_k = \prod_{i=1}^{k} \lambda_i(\mathsf{C}) \tag{3.191}$$

With an orthonormal basis H of the nullspace of C perpendicular to its rows $\mathsf{H} \perp \mathsf{C}$ this is expressible as the determinant of the regular matrix (cf. [Koch, 1997, p.65f])

$$|\mathsf{C}|_k = \left|\mathsf{C} + \mathsf{H}^T\mathsf{H}\right| \tag{3.192}$$

Obviously the multiplication of the singular matrix with a scalar results in

$$\left|\sigma^2\mathsf{C}\right|_k = \prod_{i=1}^{k} \sigma^2\lambda_i(\mathsf{C}) = \sigma^{2k}\,|\mathsf{C}|_k = \sigma^{2k}\left|\mathsf{C} + \mathsf{H}^T\mathsf{H}\right| \tag{3.193}$$

The entropy of the degenerate Normal distribution having singular covariance matrix $\sigma^2\mathsf{C}$ can be defined for the Lebesgue measure on the subspace spanned by C, so that the maximum differential entropy of distributions with this covariance matrix is given by (cf. [Kagan *et al.*, 1973, p.410] and [Cover and Thomas, 1991, p.270])

$$\begin{aligned}
h\left(\sigma^2\mathsf{C}\right) &= \frac{k}{2}(1+\log 2\pi) + \frac{1}{2}\log\left|\sigma^2\mathsf{C}\right|_k && (3.194)\\
&= \frac{k}{2}(1+\log 2\pi) + \frac{1}{2}\log\sigma^{2k}\left|\mathsf{C} + \mathsf{H}^T\mathsf{H}\right| && (3.195)\\
&= \frac{k}{2}(1+\log 2\pi) + \frac{k}{2}\log\sigma^2 + \frac{1}{2}\log\left|\mathsf{C} + \mathsf{H}^T\mathsf{H}\right| && (3.196)
\end{aligned}$$

Now the entropy change incurred by the incremental estimation is given by

$$\begin{aligned}
\Delta h &= h\left(\hat{\mathsf{C}}_{\hat{p}_2\hat{p}_2}\right) - h\left(\hat{\mathsf{C}}_{\hat{p}_1\hat{p}_1}\right) && (3.197)\\
&= h\left(\hat{\sigma}_2^2\mathsf{C}_{\hat{p}_2\hat{p}_2}\right) - h\left(\hat{\sigma}_1^2\mathsf{C}_{\hat{p}_1\hat{p}_1}\right) && (3.198)\\
&= \frac{k}{2}\log\frac{\hat{\sigma}_2^2}{\hat{\sigma}_1^2} + \frac{1}{2}\log\frac{\left|\mathsf{C}_{\hat{p}_2\hat{p}_2} + \mathsf{H}^T\mathsf{H}\right|}{\left|\mathsf{C}_{\hat{p}_1\hat{p}_1} + \mathsf{H}^T\mathsf{H}\right|} && (3.199)\\
&= \frac{k}{2}\log\frac{\hat{\sigma}_2^2}{\hat{\sigma}_1^2} + \frac{1}{2}\log\left|\left(\mathsf{C}_{\hat{p}_2\hat{p}_2} + \mathsf{H}^T\mathsf{H}\right)\left(\mathsf{C}_{\hat{p}_1\hat{p}_1} + \mathsf{H}^T\mathsf{H}\right)^{-1}\right| && (3.200)\\
&= \frac{k}{2}\log\frac{\hat{\sigma}_2^2}{\hat{\sigma}_1^2} && (3.201)
\end{aligned}$$

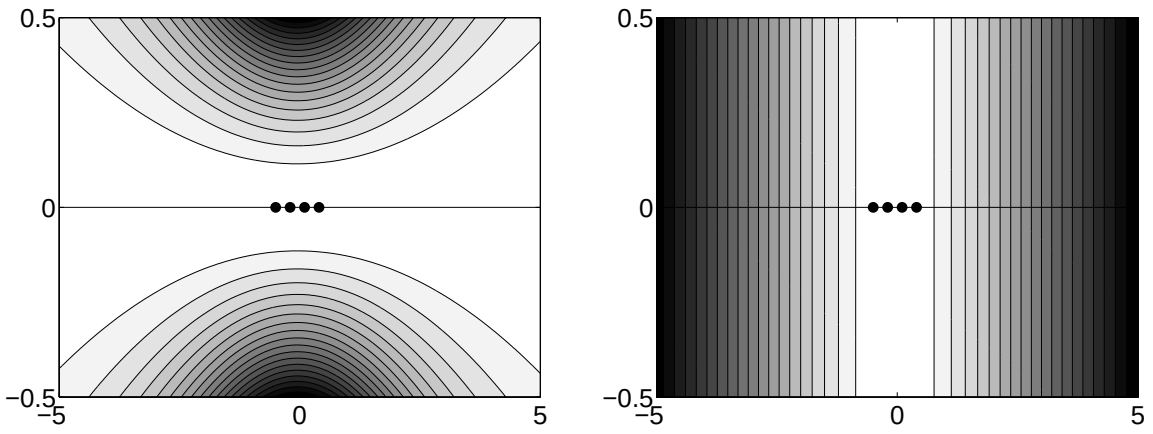

Figure 3.2: *Left:* An existing group of four 2d points with isotropic equal size covariance matrices incident to a horizontal 2d line. For each position the observation dependent entropy increase is computed that another incident 2d point would incur, and the contour lines are plotted. Points in the white region are compatible with the group model based on the observation dependent entropy increase. *Right:* The same configuration as on the left. Now the design dependent entropy decrease is plotted for each point. The proposed grouping algorithm favors points that are nearer to the existing points.

$$\begin{aligned} & + \frac{1}{2} \log \left| \left(C_{\hat{p}_1 \hat{p}_1} - F A_2 (C_{\hat{p}_1 \hat{p}_1} + H^T H) + H^T H \right) \left(C_{\hat{p}_1 \hat{p}_1} + H^T H \right)^{-1} \right| \\ = & \underbrace{\frac{k}{2} \log \frac{\hat{\sigma}_2^2}{\hat{\sigma}_1^2}}_{\Delta h_o} + \underbrace{\frac{1}{2} \log |I - F A_2|}_{-\Delta h_d} \end{aligned} \tag{3.202}$$

Note that this result is completely analogous to the result for the Gauss-Markov-Model without restrictions presented in [Beder, 2005].

Equation (3.202) shows that the entropy is increased due to the observation dependent term (see figure 3.2, left)

$$\Delta h_o = \frac{k}{2} \log \frac{\hat{\sigma}_2^2}{\hat{\sigma}_1^2} \tag{3.203}$$

that reflects, how well the new set of observations fits the current model, and simultaneously decreased due to the design dependent term (see figure 3.2, right)

$$\Delta h_d = -\frac{1}{2} \log |I - F A_2| \tag{3.204}$$

that reflects the commitment, the algorithm makes in grouping the specific set of observations. Some remarks on both quantities are in order.

First note that the observation dependent entropy increase

$$\Delta h_o = \frac{k}{2} \log F \tag{3.205}$$

is a monotonous function of the well-known Fisher-distributed test statistic (cf. [Förstner and Wrobel, 2004, p.79])

$$F = \frac{\hat{\sigma}_2^2}{\hat{\sigma}_1^2} \tag{3.206}$$

for checking, if the variance of the parameter distribution due to incorporating a new set of observation significantly increases. Hence, a new set of observations fits the existing group model, if

$$F < T_{\alpha,N_1+N_2-U+H,N_1-U+H} \tag{3.207}$$

$$\Leftrightarrow \quad \Delta h_o < \frac{k}{2} \log T_{\alpha,N_1+N_2-U+H,N_1-U+H} \tag{3.208}$$

with the threshold $T_{\alpha,m,n}$ being numerically derived from the Fisher distribution as (cf. [Koch, 1997, p.139])

$$\frac{m^{\frac{m}{2}} n^{\frac{n}{2}} \int_0^\infty t^{\frac{m}{2}+\frac{n}{2}-1} e^{-t} dt}{\int_0^\infty t^{\frac{m}{2}-1} e^{-t} dt \int_0^\infty t^{\frac{n}{2}-1} e^{-t} dt} \int_0^{T_{\alpha,m,n}} \frac{w^{\frac{m}{2}-1}}{(n+mw)^{\frac{m}{2}+\frac{n}{2}}} dw = \alpha \tag{3.209}$$

Thresholding only the observation dependent term Δh_o is conceptually equivalent to the greedy graph based approaches in section 3.4.2, because there the grouping decision is based solely on a hypothesis test on the candidate observation and the current group model. The design dependent entropy decrease is therefore not considered at all.

As mentioned, the design dependent term Δh_d reflects the commitment of the grouping algorithm that some new set of observations should be considered to belong to a certain group. In the spirit of maximum-entropy inference, this commitment should be as small as possible, so that a robust grouping algorithm makes each grouping decision, such that the design dependent entropy decrease is controlled and the maximum number of options is retained. Also observe that the expression Δh_d is closely related to the well-known hat-matrix (cf. [Förstner, 1987]), which is used to quantify the effect of gross errors in single observations on the estimation and helps to identify leverage points for the task of robust estimation.

The proposed grouping algorithm now goes as follows: starting from an initial group model derived from the set of consistent, sufficient and irreducible subsets of observations the observation dependent entropy increase and the design dependent entropy decrease are computed. From all observations, where the observation dependent entropy increase is below the threshold, i.e. the observations fit the current group model, the one with smallest design dependent entropy decrease is selected. If this design dependent entropy decrease is below some threshold, the observation is included into the group. If it is above the threshold, the current group is completed and a new group selected from the remaining set of consistent, sufficient and irreducible subsets of observations is started.

Note that the selection of fitting observations can be approximated very efficiently by checking an uncertain relation instead of thresholding the design dependent entropy increase using the data structure presented in section 2.3. The threshold on the design dependent entropy decrease is a design parameter that controls the maximum allowed distance of an observation to its group and therefore the compactness of the group. To choose this parameter one might consider that the entropy decrease of a decision should be in the same order of magnitude as the possible entropy increase caused by observed data. Therefore it is reasonable to choose the same threshold for the design dependent entropy decrease as for the observation dependent entropy increase that has been derived from the Fisher-distribution above.

The details of the proposed grouping method are shown in algorithm 7.

Up to now five different grouping methods have been presented. Apart from some remarks on specific caveats of the different methods no conclusion on their performance is possible

Algorithm 7 Grouping by bounding the entropy change

Require: observations $\mathcal{L} = \{(\boldsymbol{l}_i, \mathsf{C}_{l_i l_i})\}$, group model functions $\{\boldsymbol{g}_1, ..., \boldsymbol{g}_N\}$, constraints $\boldsymbol{h}$, threshold T_0

Ensure: group models $\hat{\mathcal{P}} = \{\hat{\boldsymbol{p}}_i\}$, assignments $\hat{\zeta}$

$\hat{\mathcal{P}} = \emptyset$
compute the set of consistent, sufficient and irreducible subsets of observation $\mathcal{S}$
while $\mathcal{S} \neq \emptyset$ **do**
  pick a subset of observations $s \in \mathcal{S}$
  compute the corresponding group model $(\hat{\boldsymbol{p}}, \hat{\mathsf{C}}_{pp})$ from s
  repeat
    $\mathcal{O} = \emptyset$
    for each observation $(\boldsymbol{l}_i, \mathsf{C}_{l_i l_i}) \in \mathcal{L} \backslash s$ **do**
      compute observation dependent entropy increase $\Delta h_o^{(i)}$ according to equation (3.203)
      if $\Delta h_o < \frac{k}{2} \log \frac{1}{T_{\alpha, N_1+N_2-U+H, N_1-U+H}}$ **then**
        compute design dependent entropy decrease $\Delta h_d^{(i)}$ according to equation (3.204)
        if $\Delta h_d < T_0 + \frac{k}{2} \log \frac{1}{T_{\alpha, N_1+N_2-U+H, N_1-U+H}}$ **then**
          $\mathcal{O} = \mathcal{O} \cup \{(\boldsymbol{l}_i, \mathsf{C}_{l_i l_i})\}$
        end if
      end if
    end for
    $s = s \cup \left\{ \text{argmin}_{(\boldsymbol{l}_i, \mathsf{C}_{l_i l_i}) \in \mathcal{O}} \left(\Delta h_d^{(i)} \right) \right\}$
  until $\mathcal{O} = \emptyset$
  $\mathcal{L} = \mathcal{L} \backslash s$
  $\mathcal{S} = \mathcal{S} \backslash \{t \in \mathcal{S} | s \cap t \neq \emptyset\}$
  $\hat{\mathcal{P}} = \hat{\mathcal{P}} \cup \{\hat{\boldsymbol{p}}(s)\}$
  assign all elements in s to $\hat{\boldsymbol{p}}(s)$
end while

so far. In the following the performance of those five different methods on a synthetic data set will therefore be analyzed and compared. Furthermore, building reconstruction results obtained using the graph based clique partition grouping method and the entropy bound grouping method will be demonstrated in chapter 4.

3.5 Empirical analysis on synthetic data

In order to compare the performance of the presented grouping methods, all algorithms were run on a simple synthetic dataset, where ground truth is available. The dataset was generated as follows: first K line segments were generated at random inside the box $[-1, ..1] \times [-1, ..., 1]$. Then for each such line segment a set of $\frac{N}{K}$ random points on the segment were generated and perturbed by Gaussian noise with standard deviation σ yielding a total of N inlier points. Finally a number N_{out} of outliers were generated at random inside the box $[-1.5, ..1.5] \times [-1.5, ..., 1.5]$. Hence, the resulting dataset is dependent on the following parameters:

- K is the number of groups
- $\frac{N}{K}$ is the number of observations per group
- N_{out} is the number of outliers
- σ is the standard deviation of the noise of the observations

The effect of all those parameters on the classification results obtained by the five different grouping algorithms will be studied in the following.

The points were grouped together based on line incidence as group model function using the expectation maximization method, the sampling consensus method, the connected components and clique partition method and the entropy bound method presented in the previous sections. As the EM method is unable to find the number of groups itself, the effect of a wrongly set parameter is studied by comparing also the grouping results using the correct number of groups as well as with one group less and one group more. This yields a total of seven different grouping algorithms, that are abbreviated in the figures as follows:

- EM-1 is the expectation maximization method with one group less than correct
- EM is the expectation maximization method with the correct group size
- EM+1 is the expectation maximization method with one group more than correct
- CC is the graph based connected component method
- Cl is the graph based clique partition method
- Smp is the sampling consensus method
- Ent it the entropy bound method

The computed group memberships were compared to the known ground truth memberships used to generate the data and the performance was assessed using the percentage of correctly classified observations with respect to the ground truth without counting the outliers. The experiment was repeated many times and the mean classification rates together with their empirical standard deviation were derived for different input parameters. The results are depicted in the figures and will be discussed in the following.

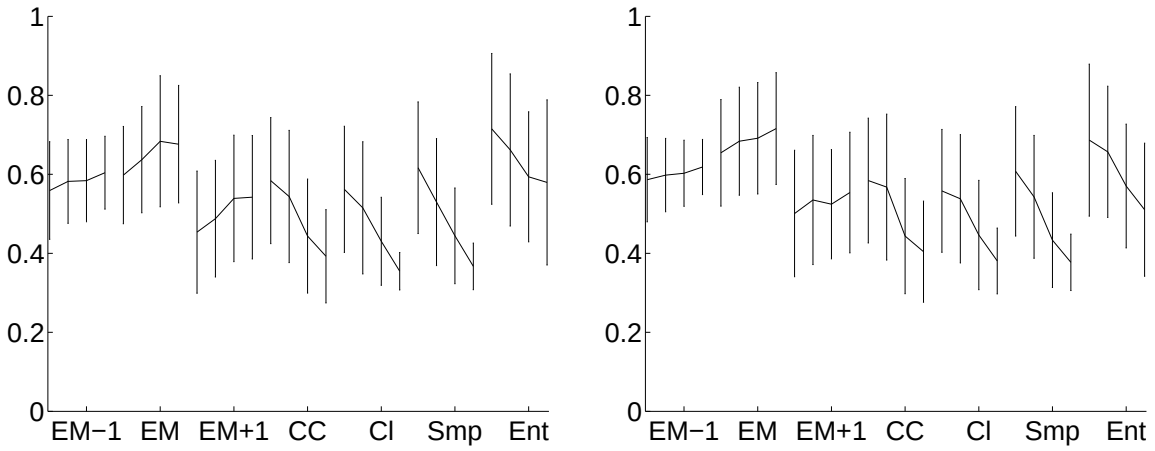

Figure 3.3: *Left:* Effect of the group size on the performance. Classification rates (with standard deviations) for $K = 3$ groups with group sizes $\frac{N}{K} = 3, 5, 10, 20$, point accuracy $\sigma = 0.01$ and number of outliers $N_{out} = 0$ for the different algorithms. Observe that the classification rates increase for the EM-algorithm and decrease for the other algorithms. For small group sizes, the entropy bound approach is slightly better than all other methods. *Right:* The same as on the left hand side with the number of outliers being equal to the group size, i.e. $N_{out} = \frac{N}{K}$. Note, the effect of outliers is negligible.

First the effect of the group size $\frac{N}{K}$ on the grouping performance was analyzed. Therefore the number of groups was set to $K = 3$ and the point accuracy to $\sigma = 0.01$. On the left hand side of figure 3.3 the performance results for group sizes $\frac{N}{K} = 3, 5, 10$ and 20 are shown with no outliers, i.e. $N_{out} = 0$, for the different algorithms. On the right hand side of figure 3.3 the performance results for the same group sizes but now with the number of outliers being equal to the number of group members, i.e. $N_{out} = \frac{N}{K}$, are shown. It can be observed that the expectation maximization method performs the better, the more observations per group are given. This is in contrast to the other methods, which decrease in classification performance, if more observations per group are present. The reason for this is that the number of identified groups increases with more observations for each method that does not fix K, which the expectation maximization method does. Also observe that the entropy bound method achieves comparable results for small group sizes to the expectation maximization method with correctly chosen number of groups K. It is slightly superior to the expectation maximization method, if the number of groups used in the expectation maximization method is incorrectly estimated as $K+1$ or $K-1$, though. The graph based methods and the sampling consensus methods yield very similar results, which are slightly inferior to the entropy bound approach. The presence of a number of outliers equal to the group size does not alter the picture, as very similar results are obtained.

To quantify the effect of a reasonable amount of outliers on the classification performance, the experiment was run again using a group size of $\frac{N}{K} = 5$ and an accuracy of $\sigma = 0.1$. On the left hand side of figure 3.4 the classification rates for $K = 3$ groups with the number of outliers being $N_{out} = 0, 5, 10, 15$ are shown for each algorithm. On the right hand side of figure 3.4 the classification rates for $K = 5$ groups with the number of outliers being $N_{out} = 0, 5, 10, 25$ are shown. It can be observed that, as in the previous experiment, the number of outliers has no significant effect on the grouping results, as all algorithms are designed to work robustly. The classification performance of the entropy bound method is again slightly superior to the

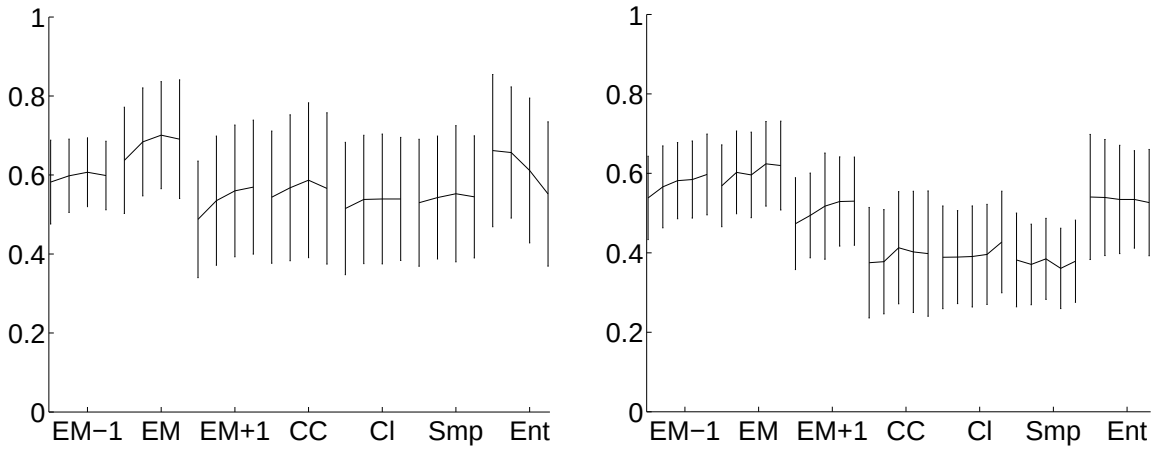

Figure 3.4: *Left:* Effect of outliers on the performance. Classification rates (with standard deviations) for $K = 3$ groups with group size $\frac{N}{K} = 5$, point accuracy $\sigma = 0.01$ and number of outliers $N_{out} = 0, 5, 10, 15$ for the different algorithms. Observe that the effect of a reasonable number of outliers is negligible. *Right:* Classification rates (with standard deviations) for more groups $K = 5$ than on the left hand side with group size $\frac{N}{K} = 5$, point accuracy $\sigma = 0.01$ and number of outliers $N_{out} = 0, 5, 10, 20, 25$ for the different algorithms. Again the effect of a reasonable number of outliers on the results is negligible.

other approaches.

Next the effect of the group members accuracy on the classification performance was studied. The number of groups was chosen as $K = 3$ and the observations per group as $\frac{N}{K} = 5$. On the left hand side of figure 3.5 the classification rates for accuracies in the range $\sigma = 0.001, 0.05, 0.01, 0.1, 0.2$ are shown with no outliers present. On the right hand side of figure 3.5 the classification results for the same accuracies with a number $N_{out} = 10$ outliers present is shown. It can be observed that the classification rates drop, if the point accuracy decreases. This is due to the fact that the group memberships become weaker and the uncertainty which observation belongs to which group increases. Also note that for extremely accurate points the classification rates of the graph based methods, the sampling consensus method and the entropy bound method are comparable and slightly superior to the expectation maximization approach. For large uncertainties the graph based methods and the sampling consensus method become useless, while the entropy bound method is still applicable in this case. The effect of accuracy on the classification rates of the expectation maximization method is not as significant as with the other methods. Again the effect of outliers is negligible.

Finally the effect of the number of groups on the classification rates was analyzed. The number of observations per group was set to $\frac{N}{K} = 5$ and the accuracy of the points to $\sigma = 0.01$. On the left hand side of figure 3.6 the classification performance is show for the number of groups being $K = 2, 3, 5$ without outliers and on the right hand side of figure 3.6 the classification performance is shown for the same configuration but with the number of outliers being $N_{out} = 10$. It can be observed that the classification rates decrease with increasing number of groups. This is caused on the one hand by a higher chance of misclassification and on the other hand by a higher chance of ambiguous configurations such as intersecting line segments, if more of them are present. Again, the graph based methods and the sampling consensus method perform equally well, while the entropy bound method performs slightly

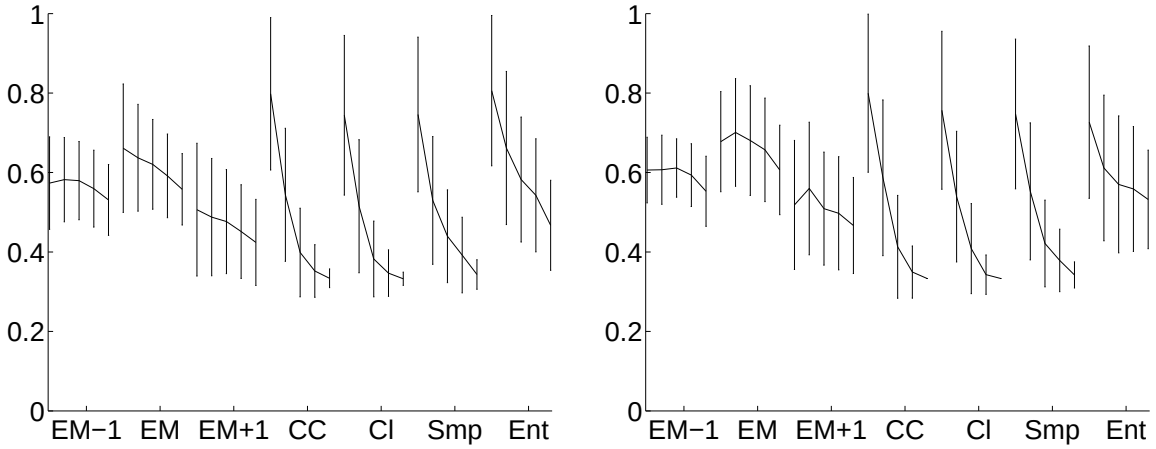

Figure 3.5: *Left:* Effect of points accuracy on the performance. Classification rates (with standard deviations) for $K = 3$ groups with group size $\frac{N}{K} = 5$ and point accuracy $\sigma = 0.001, 0.05, 0.01, 0.1, 0.2$ without outliers for the different algorithms. Note that all algorithms perform well for $\sigma = 0.001$ and that the entropy bound approach performs slightly better than the rest for larger values of σ. Also observe that the effect of accuracy on the expectation maximization method is small. *Right:* The same as on the left hand side with the number of outliers being $N_{out} = 10$. Note that the effect of the outliers is negligible.

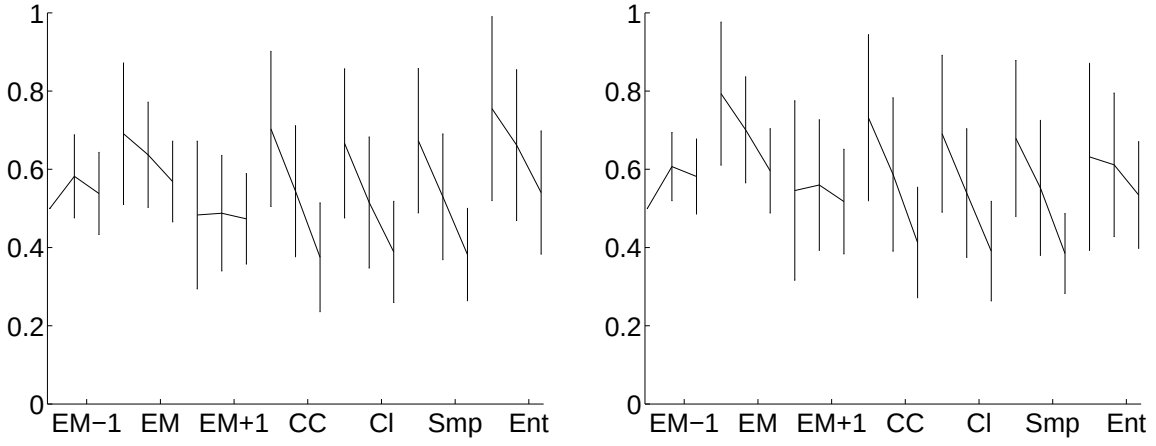

Figure 3.6: *Left:* Effect of the number of groups on the performance. Classification rates (with standard deviations) for $K = 2, 3, 5$ groups with group size $\frac{N}{K} = 5$ and point accuracy $\sigma = 0.01$ without outliers for the different algorithms. Observe that the classification rates decrease with the number of groups and the the entropy bound approach is slightly better than the other methods. *Right:* The same as on the left hand side with the number of outliers being $N_{out} = 10$. Again, the presence of outliers has no significant effect.

better than the others. Also note that the performance decrease for the entropy bound method is slower. Again the effect of a small number of outliers is negligible.

This section demonstrated the performance of the presented grouping methods on a simple synthetic data set under a range of varying conditions. The availability of ground truth for the synthetic test data enabled the quantitative comparison of the different methods. In the next chapter the applicability of the proposed grouping methods for the task of fully automatic building reconstruction from aerial images will be demonstrated. As no ground truth is available in this case, the results will be judged only visually there.

Chapter 4

Application to building reconstruction

4.1 Input data set and task

In the following the applicability of the presented geometric grouping methods for the task of building reconstruction will be demonstrated. The presented reconstruction results including the matching of feature points and lines are solely based on the geometry of the images rather than their radiometric content. This is in order to show the power of the geometric cues in conjunction with the grouping methodology, as it will be seen that matching between image points and image lines is indeed possible without using radiometric information from the images at all. In practical applications the radiometric information contained in the images should be used, though. However, this has not been the scope of this study.

To demonstrate the applicability of the presented geometric grouping tasks and methods for building reconstruction from aerial images a data set of digital aerial images with high resolution and high overlap taken in a suburban area in Toyonaka (Japan) has been used. The pictures were provided by the company Vexcel and taken using the airborne digital camera UltraCam-D. The camera calibration for this specific camera results from the stitching based digital process of picture generation (cf. [Kröpfl *et al.*, 2004]) and is therefore provided by the manufacturer. The exterior orientation has been established by a bundle block adjustment over 50 images in 5 strips.

In figures 4.1,4.3,4.5 and 4.7 twelve pictures of single buildings cut out of the large aerial images are shown. Figure 4.1 shows simple and small buildings, figure 4.3 shows simple and larger buildings. Figure 4.5 shows more complex but small buildings and figure 4.7 shows complex and large buildings.

Those buildings were cut out of each image in the block they occurred in. Point and line features together with their uncertainties were extracted from all those images using the method presented in [Fuchs, 1998]. The corresponding results of this feature extraction for the exemplary images are shown in figures 4.2, 4.4, 4.6 and 4.8. Note that in order to obtain comparability between the results, only one parameter set was used for all experiments.

4.2 Grouping results

The extracted image point features and image line features were back-projected into 3d space using the known calibration and orientation as described in sections 2.1.5 and 2.1.5.

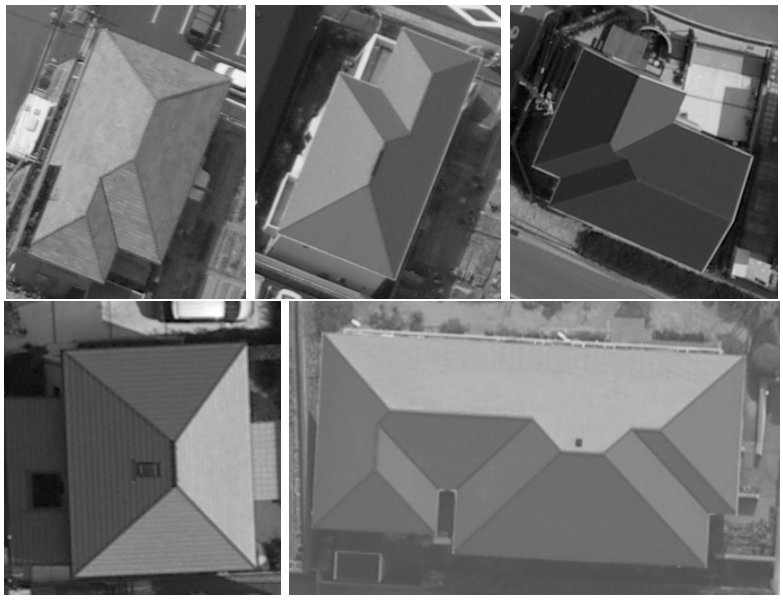

Figure 4.1: Digital aerial images of some simple small houses in a suburban area in Toyonaka (Japan).

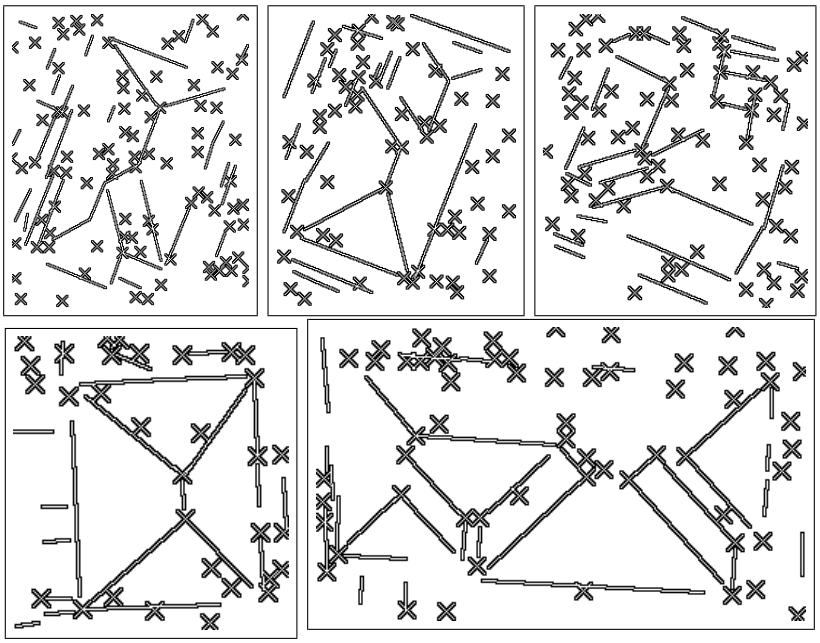

Figure 4.2: Point and line features extracted from the images depicted in figure 4.1.

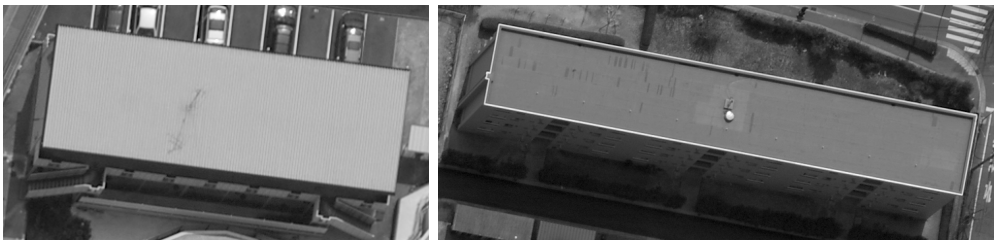

Figure 4.3: Digital aerial images of some simple large houses in a suburban area in Toyonaka (Japan).

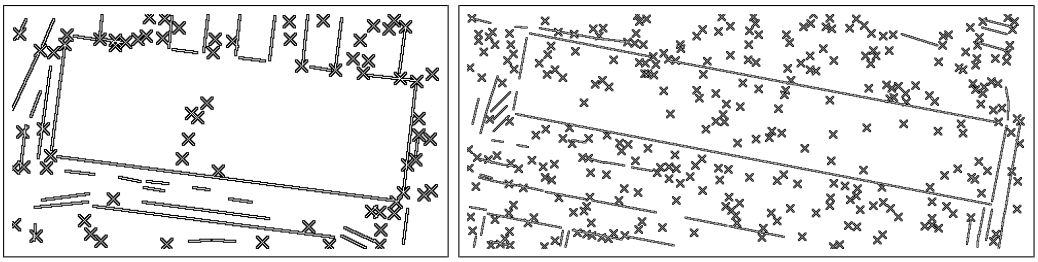

Figure 4.4: Point and line features extracted from the images depicted in figure 4.3.

Figure 4.5: Digital aerial images of some complex small houses in a suburban area in Toyonaka (Japan).

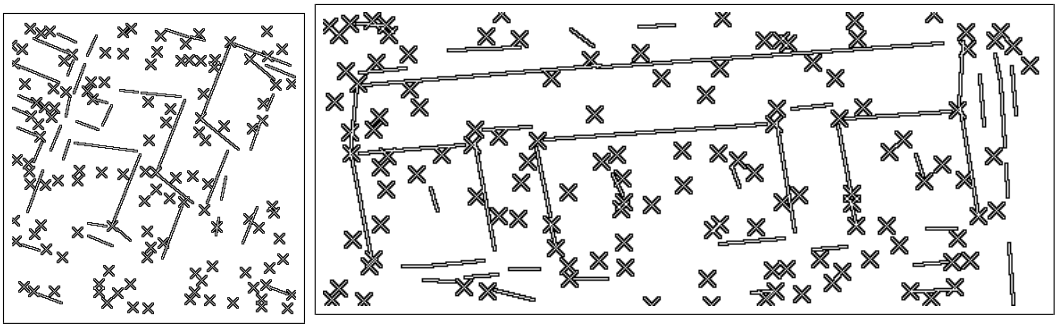

Figure 4.6: Point and line features extracted from the images depicted in figure 4.5.

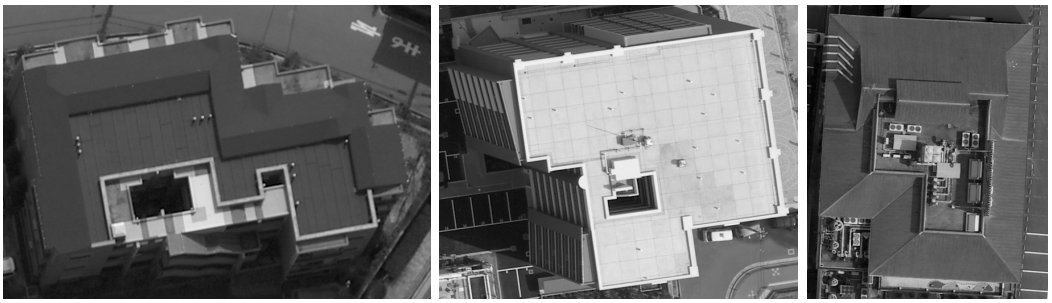

Figure 4.7: Digital aerial images of some complex large houses in a suburban area in Toyonaka (Japan).

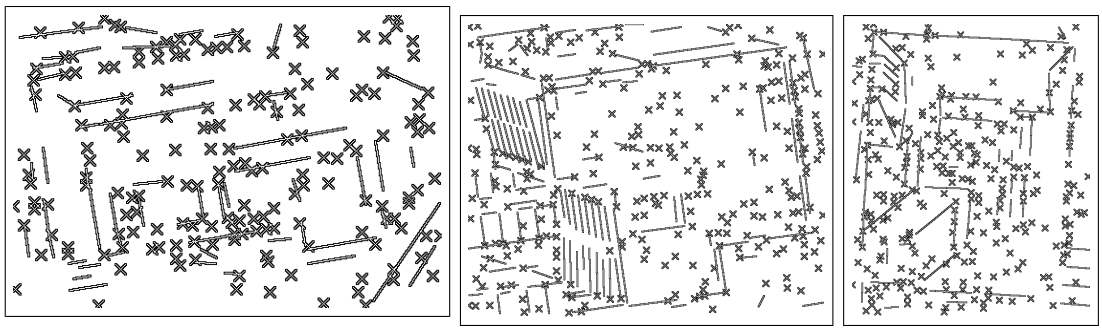

Figure 4.8: Point and line features extracted from the images depicted in figure 4.7.

Hence, every image point corresponds to an uncertain oriented projective 3d line segment and every image line segment corresponds to a convex uncertain oriented projective 3d polygon. The resulting uncertain oriented projective 3d line segments were grouped into 3d points as outlined in section 3.3.3 and the uncertain oriented projective 3d polygons were grouped into 3d lines as outlined in section 3.3.4. For each of the twelve houses in the dataset this grouping has been established using the clique partition method outlined in section 3.4.2 and the entropy bound method outlined in section 3.4.4. Trivial groups of size smaller than three were discarded from the results. Also note that the parameters of the algorithms were not changed between the different scenarios.

In figure 4.9 the reconstruction results of a small simple house roof from a strong geometric configuration of 19 images are shown. The roof of this house is slightly textured, resulting in a large number of identifiable and matched points. As single observations are allowed to be assigned to multiple groups in the graph based grouping approaches, the number of groups is much larger there than in the entropy bound approach. The graph based approach missed one roof line, while the entropy bound approach recovered every meaningful line in this case. There is some obvious clutter in the reconstruction from the entropy bound approach, though the number of reconstructed elements in the graph based method is larger and therefore suggests that there is in fact a large number of very small and not very useful line segments.

In figure 4.10 the reconstruction results from 19 images for a very similar house are shown. The results look quite similar. The difference is that there is much less texture present on the roof resulting in much fewer reconstructed 3d points. Both algorithms missed a part of the roof, were the house has an adjacent structure, which is only visible in part of the images and has therefore much less support. This must be considered as a major caveat of the grouping

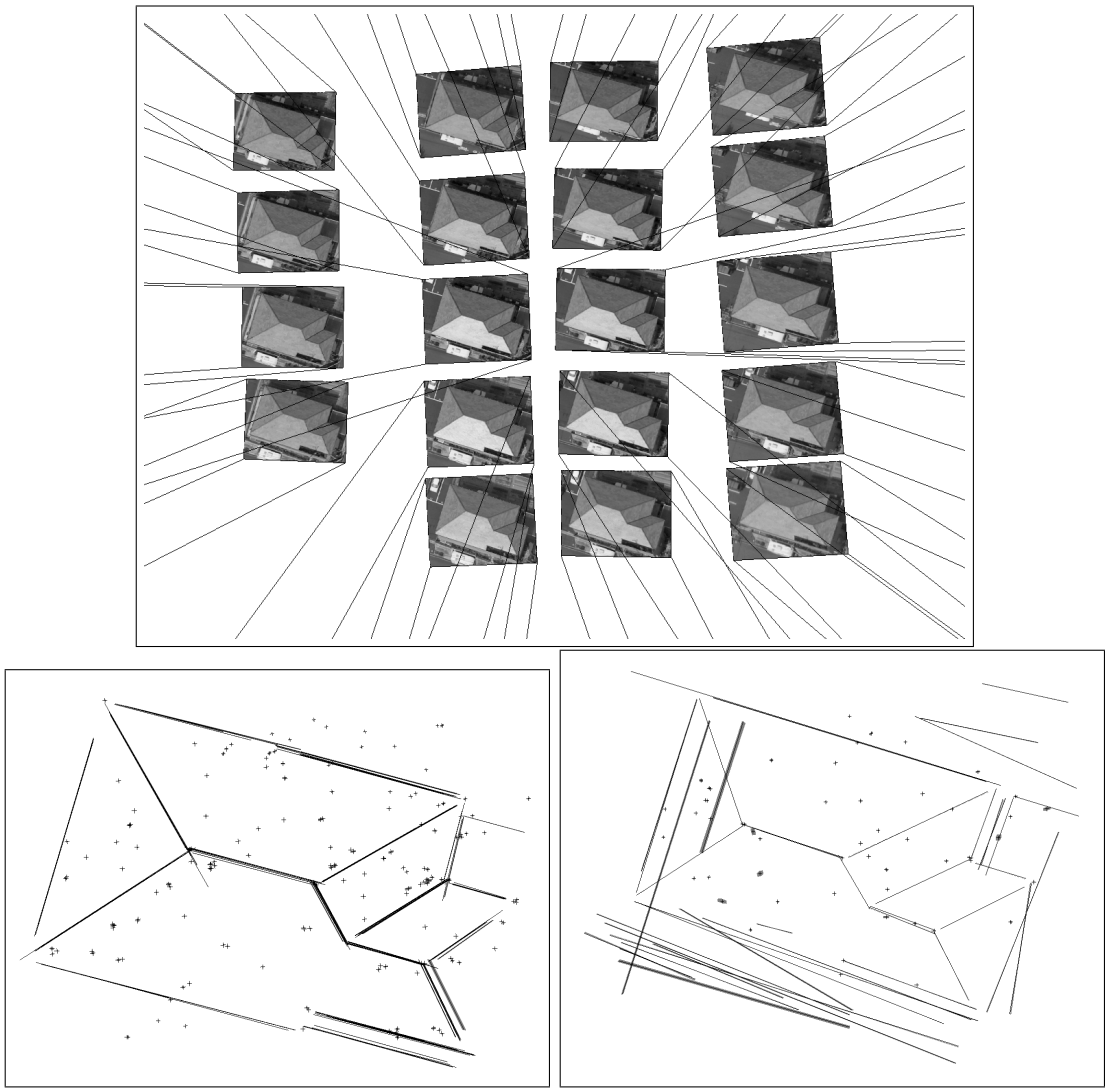

Figure 4.9: *Top:* Geometric 3d configuration of the 19 images taken form a small simple house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 162 3d points and 119 3d lines. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 54 3d points and 42 3d lines.

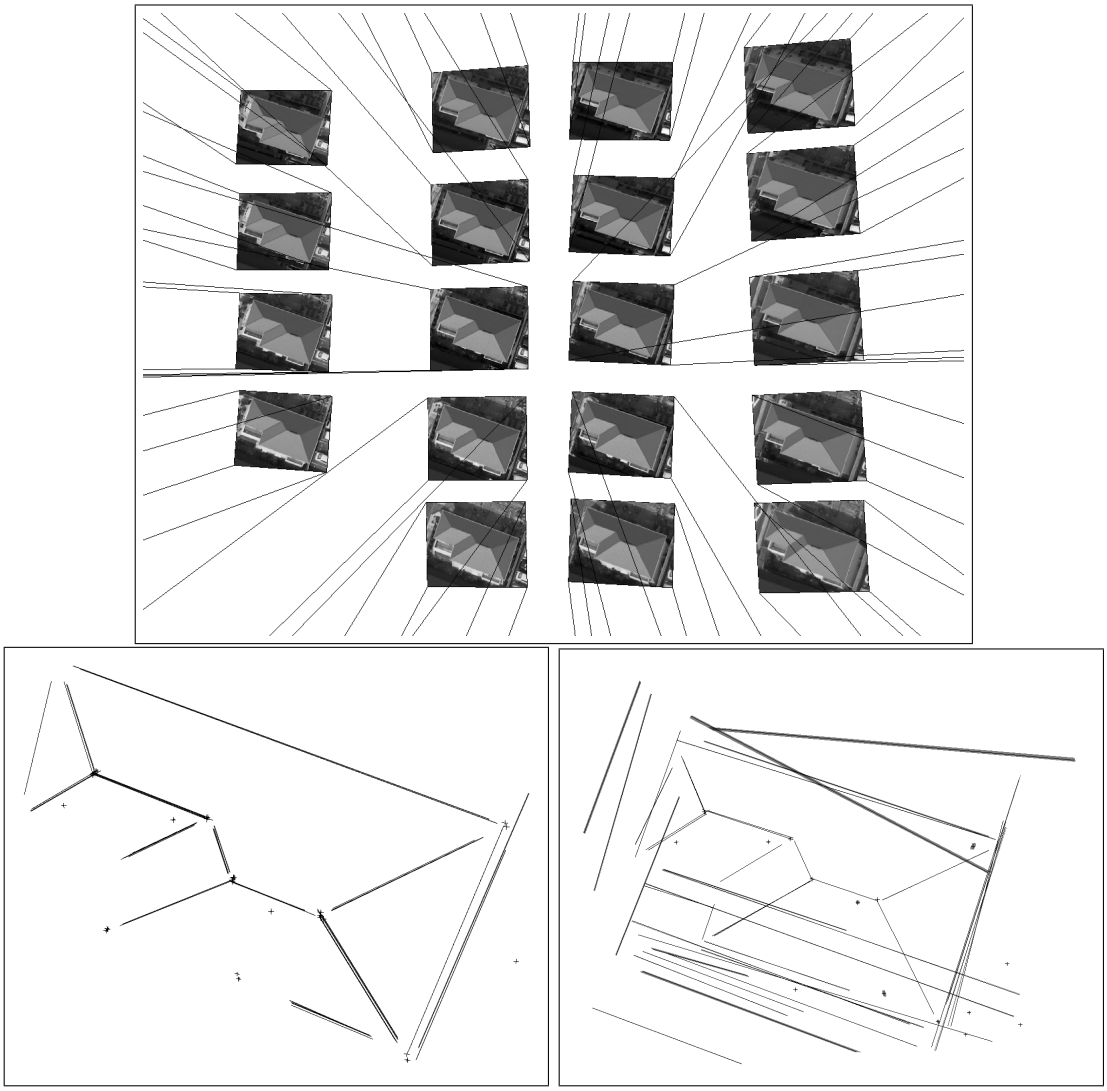

Figure 4.10: *Top:* Geometric 3d configuration of the 19 images taken form a small simple house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 28 3d points and 44 3d lines. One line is missed, where the house has an adjacent structure. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 16 3d points and 41 3d lines.

based building reconstruction, as a large number of supporting evidence is required for each scene feature and the support can vary significantly in a scene due to occlusions.

In figure 4.11 another reconstruction result for a small simple house is shown. In this case fewer images (15) are present. More important edges are missed here, though the entropy bound method performs slightly better.

In figure 4.12 even fewer images (13) are present. The entropy bound method still recovers all the edges of the roof structure correctly, while the clique partition method breaks down.

The reconstruction of a small simple house depicted in figure 4.13 is based only on six images. The clique partition approach only recovers one single line, while the entropy bound method still finds some lines on the roof. However, matching image features based on solely geometric grouping must be considered as not applicable, if only few images of the object to be reconstructed are given.

In figure 4.14 the reconstruction of a larger and very simple house roof from 16 images is shown. The clique partition approach misses the one large line on the front of the roof, which the feature extraction could only extract in small fragments due to occluding boundaries with the parking lots in front of the building. The entropy bound approach was able to group those fragments together correctly, as it incrementally builds the group model and is therefore capable of chaining observations in a controlled manner. This must be considered as one of the key features of the approach.

Figure 4.15 shows the reconstruction of a large and seemingly simple house roof from 16 images. The reconstruction results are unsatisfactory in this case for both methods. The reason could be assumed to be the feature extraction, which has great difficulties with such flat roof buildings for the chosen overall set of parameters, because lines on the delimiting wall of the roof are extracted ambiguously on its outside, on its inside at the top as well as on the inside on the bottom or on the cast shadows. Those ambiguities, resulting from difference in viewpoint and lighting conditions, yield image features, which are not co-linear in space due to the unclear semantic of the roof delineation itself, which is of course unknown to the feature extraction procedure. The effect is worsened by the non-maximum suppression of the feature extraction that often results in only one of the different possible meaningful lines to be extracted. However, the entropy bound method yields much more matching hypothesis than the clique partition method does.

Also note that the feature extraction could of course be tuned to work better for this specific building. As mentioned before, the same feature extraction has been applied for the whole dataset, in order to be able to compare the results and to demonstrate the potentials for the full automation of the procedure.

In figure 4.16 the reconstruction results for a small but complex house from 16 images are shown. While the entropy bound grouping approach finds most of the important roof structure, the clique partition method is only able to find the most dominant structures. As before, the reason for this is that complex structures yield more occlusions and therefore less geometric evidence.

In figure 4.17 the reconstruction of a small complex house from only ten images is shown. The clique partition method is not able to reconstruct one single line, while the entropy bound grouping method is again able to recover the dominant structures even from as few as ten observations.

This behavior can also be observed in the larger and complex buildings, where the reconstructions are depicted in figures 4.18, 4.19 and 4.20. While the clique partition grouping is very conservative and yields only few but correct results, the entropy bound grouping method yields better roof reconstructions that are cluttered by more outliers.

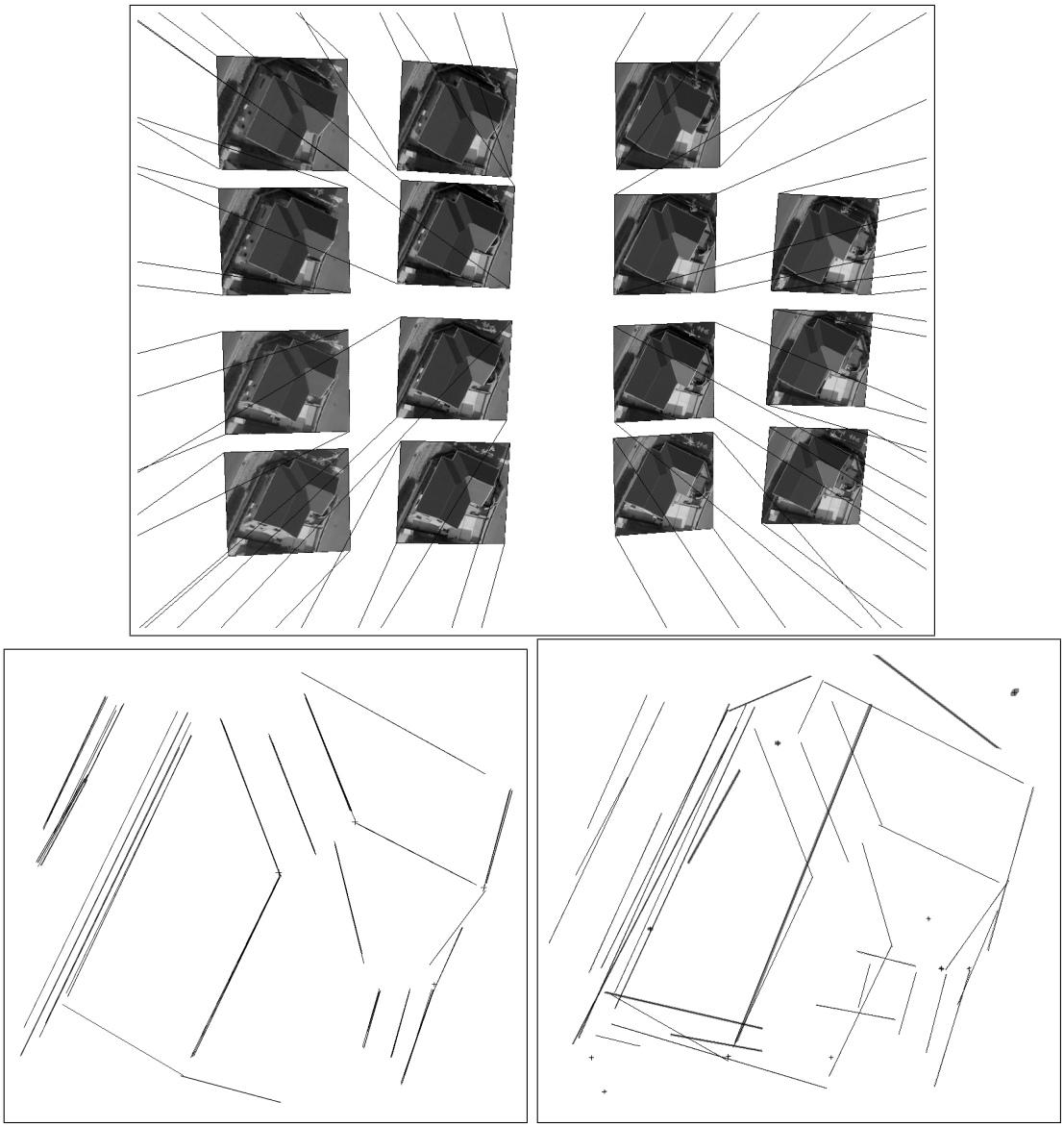

Figure 4.11: *Top:* Geometric 3d configuration of the 15 images taken form a small simple house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 4 3d points and 76 3d lines. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 10 3d points and 36 3d lines.

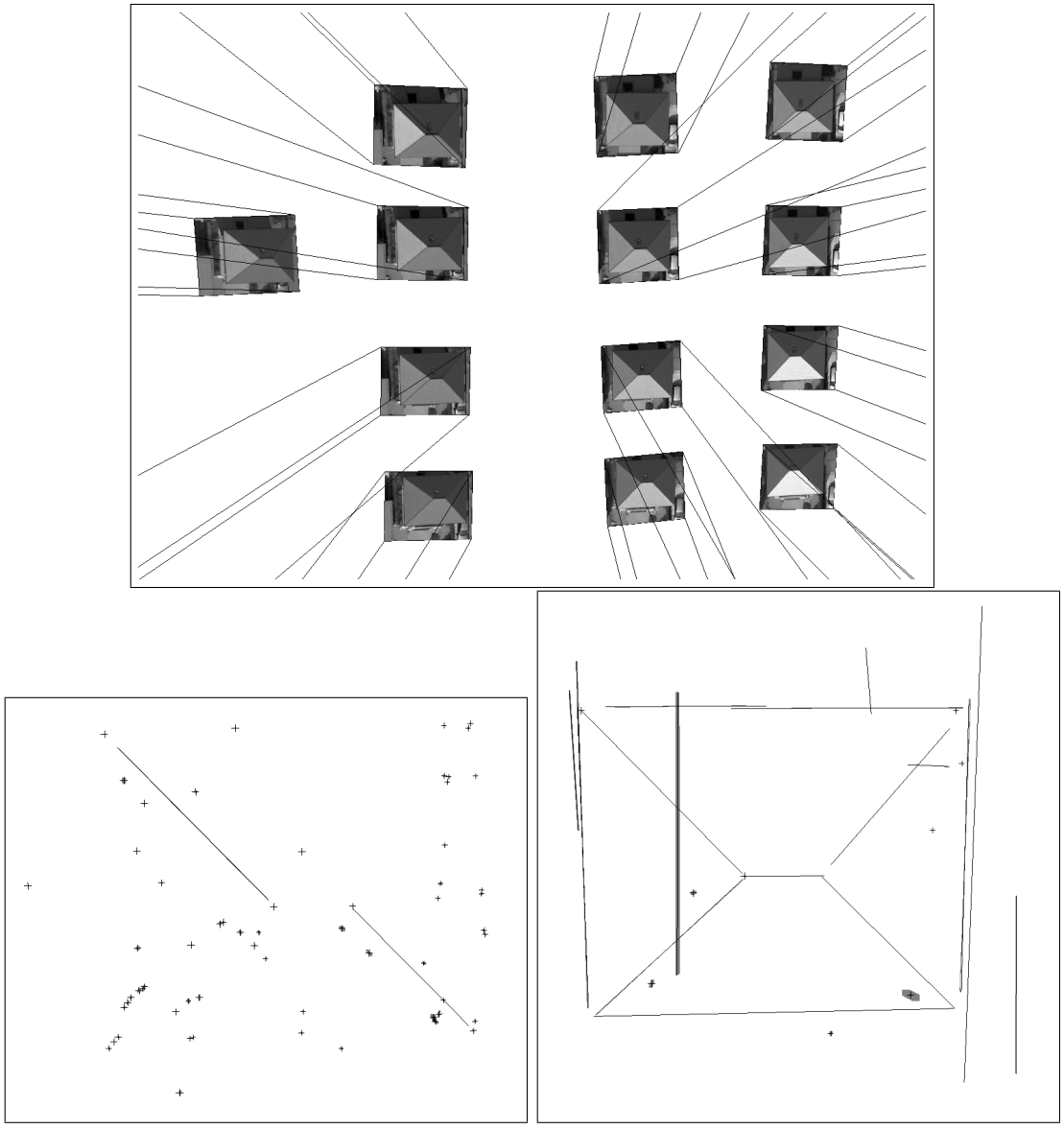

Figure 4.12: *Top:* Geometric 3d configuration of the 13 images taken form a small simple house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 61 3d points and 3 3d lines. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 9 3d points and 17 3d lines. In contrast to the clique partition method, the entropy bound method found all the important lines.

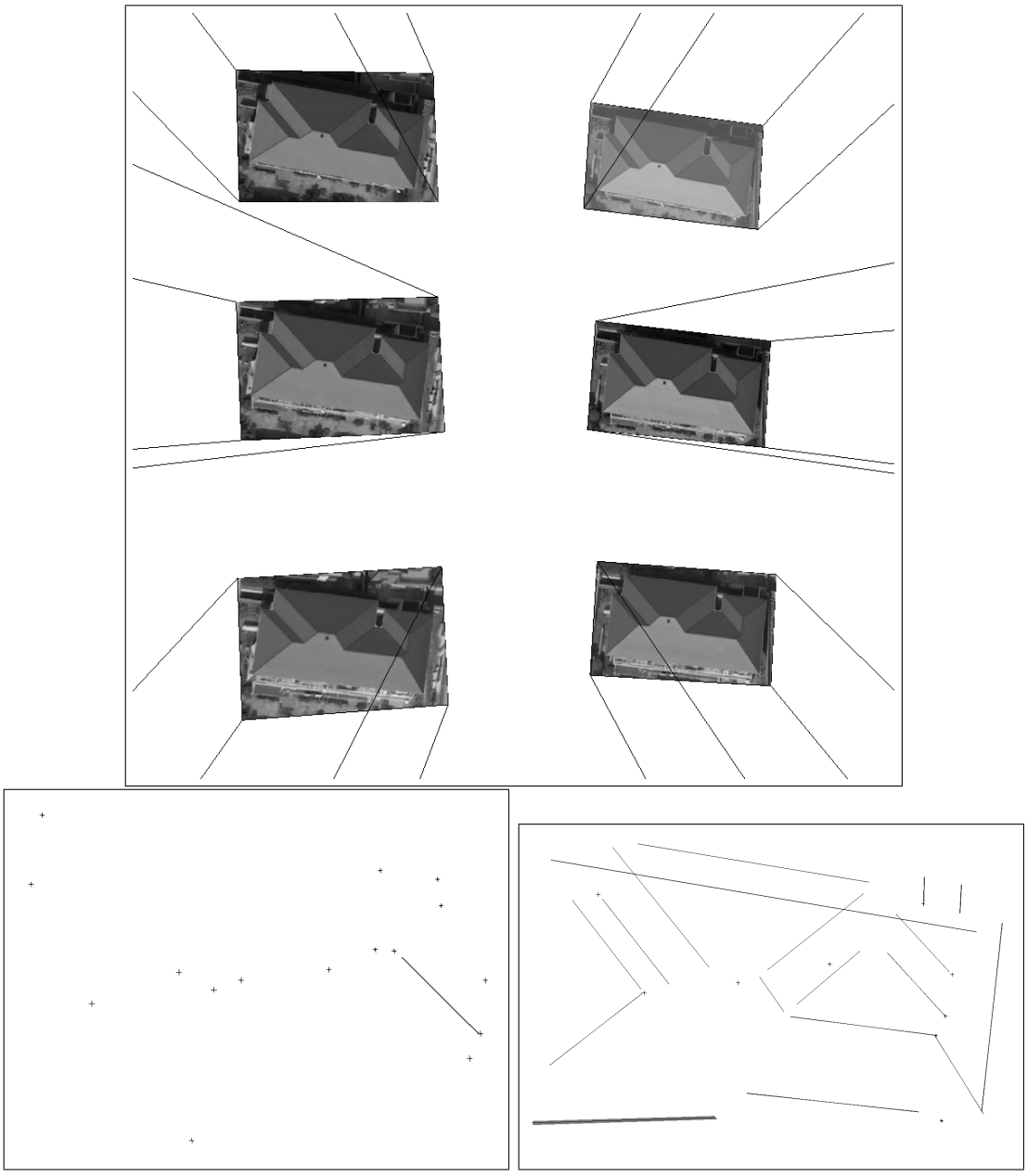

Figure 4.13: *Top:* Geometric 3d configuration of the 6 images taken form a small simple house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 18 3d points and 1 3d line. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 9 3d points and 18 3d lines.

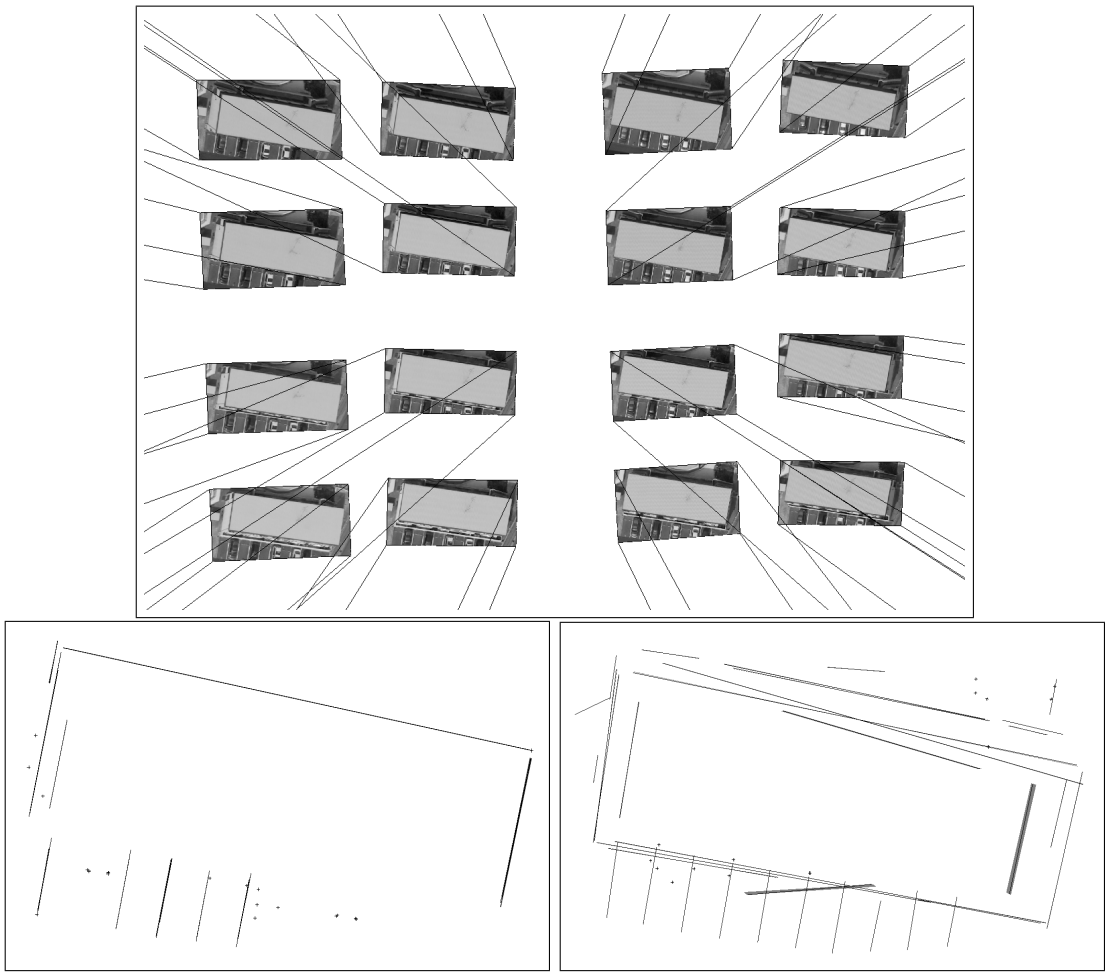

Figure 4.14: *Top:* Geometric 3d configuration of the 16 images taken form a large simple house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 21 3d points and 40 3d lines. The front roof line is not found, due to fragmented image features resulting from occluding boundaries with the parking lots. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 14 3d points and 34 3d lines.

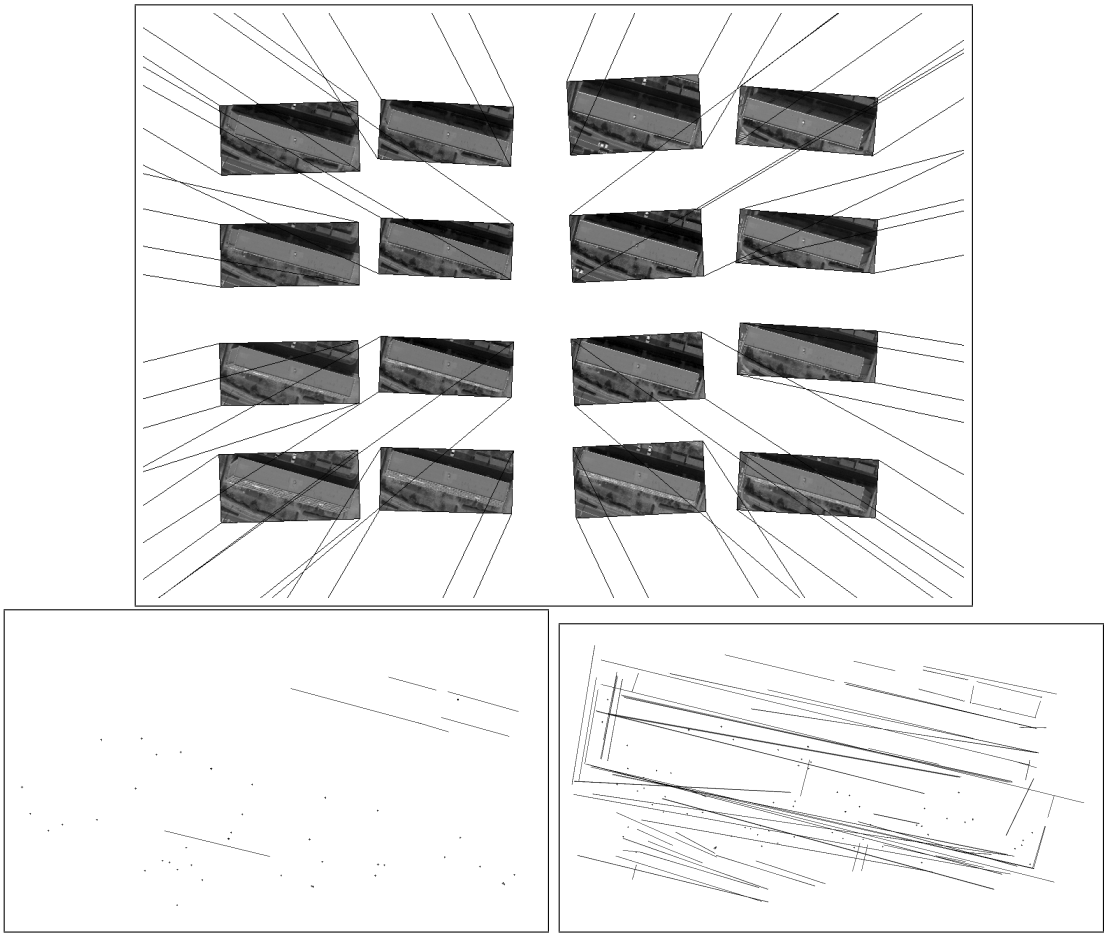

Figure 4.15: *Top:* Geometric 3d configuration of the 16 images taken form a large simple house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 45 3d points and 6 3d lines. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 69 3d points and 67 3d lines.

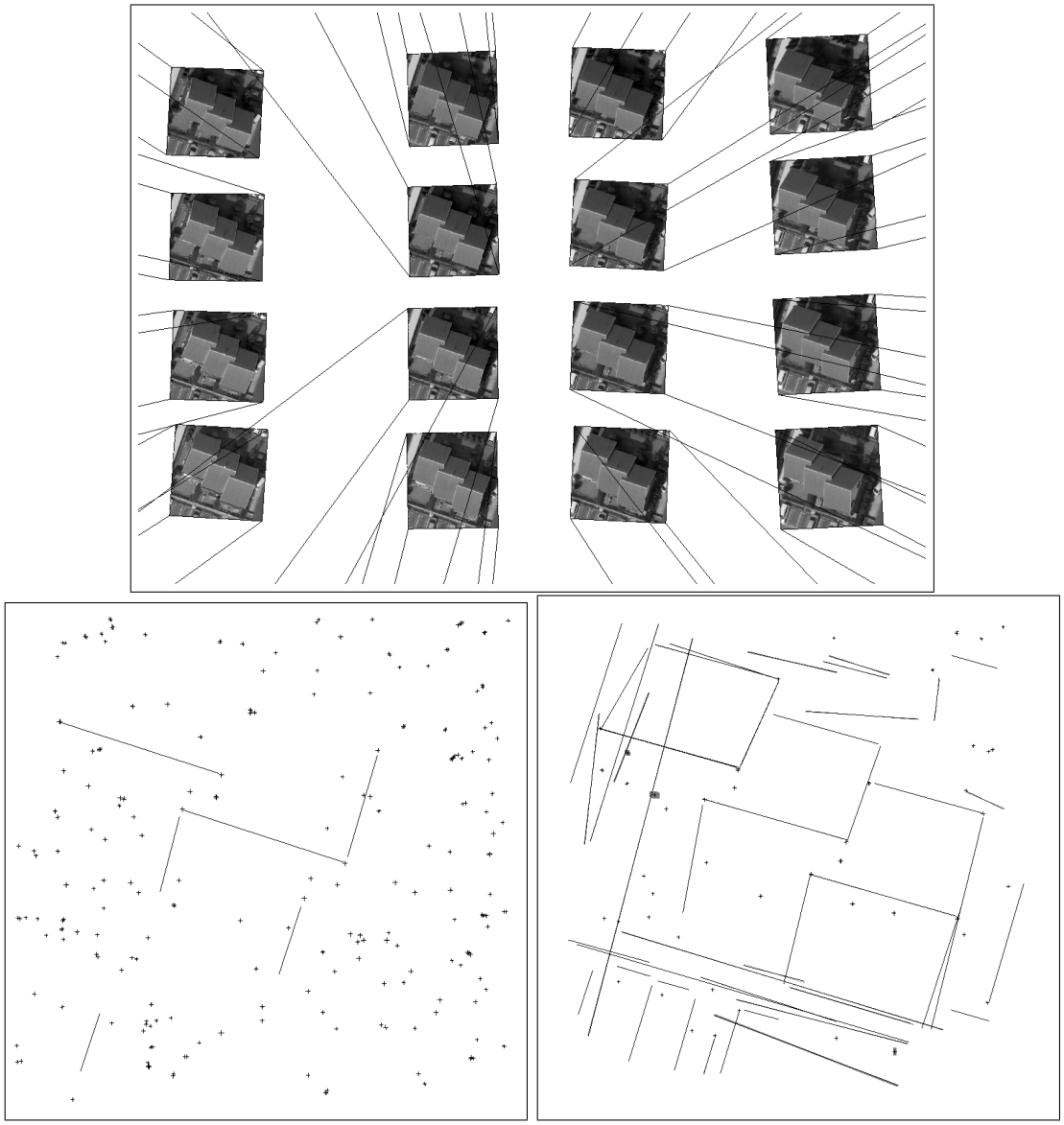

Figure 4.16: *Top:* Geometric 3d configuration of the 16 images taken form a small complex house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 203 3d points and 6 3d lines. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 51 3d points and 46 3d lines.

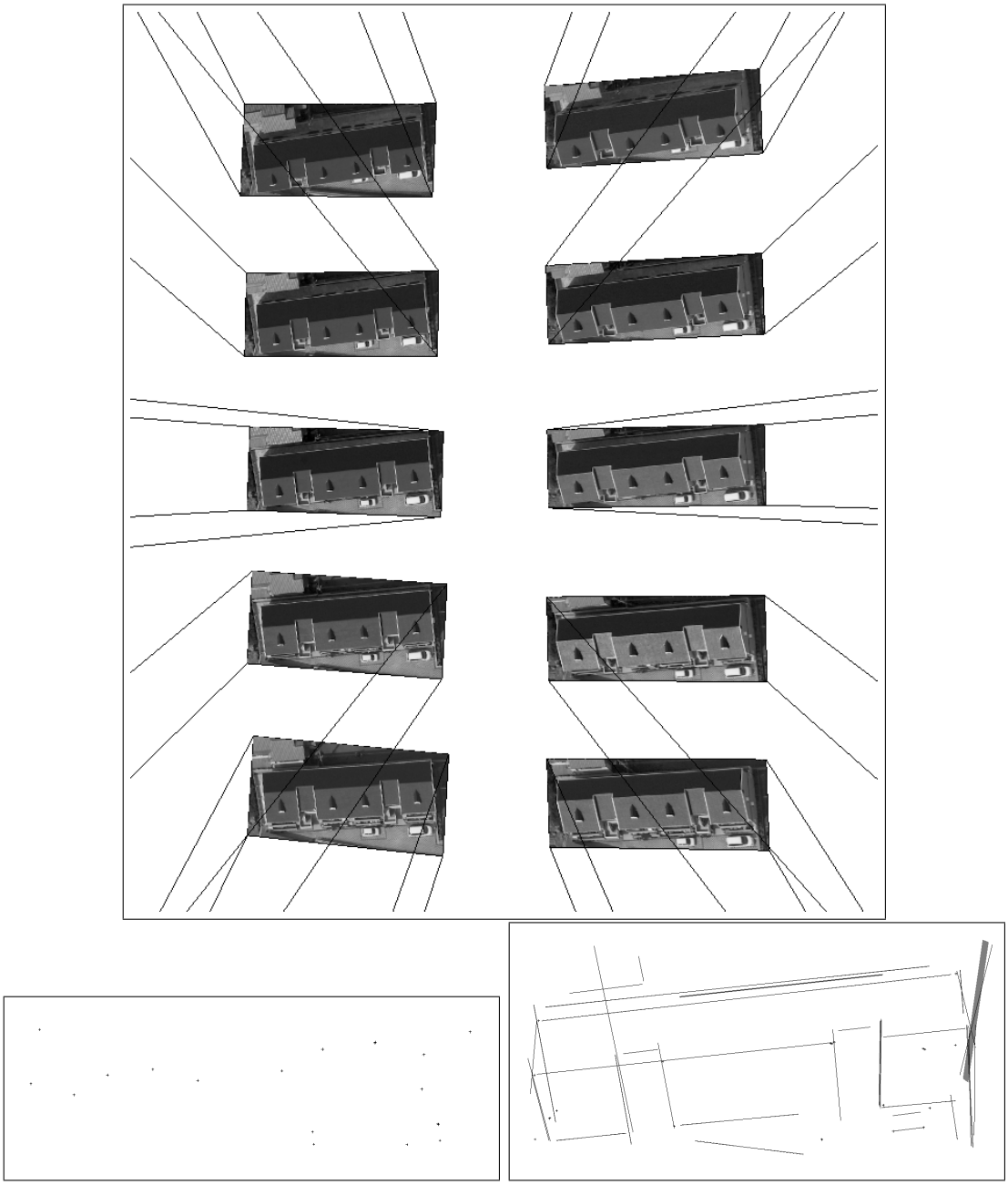

Figure 4.17: *Top:* Geometric configuration of the 10 images taken form a small complex house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 19 3d points and no 3d lines. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 17 3d points and 31 3d lines.

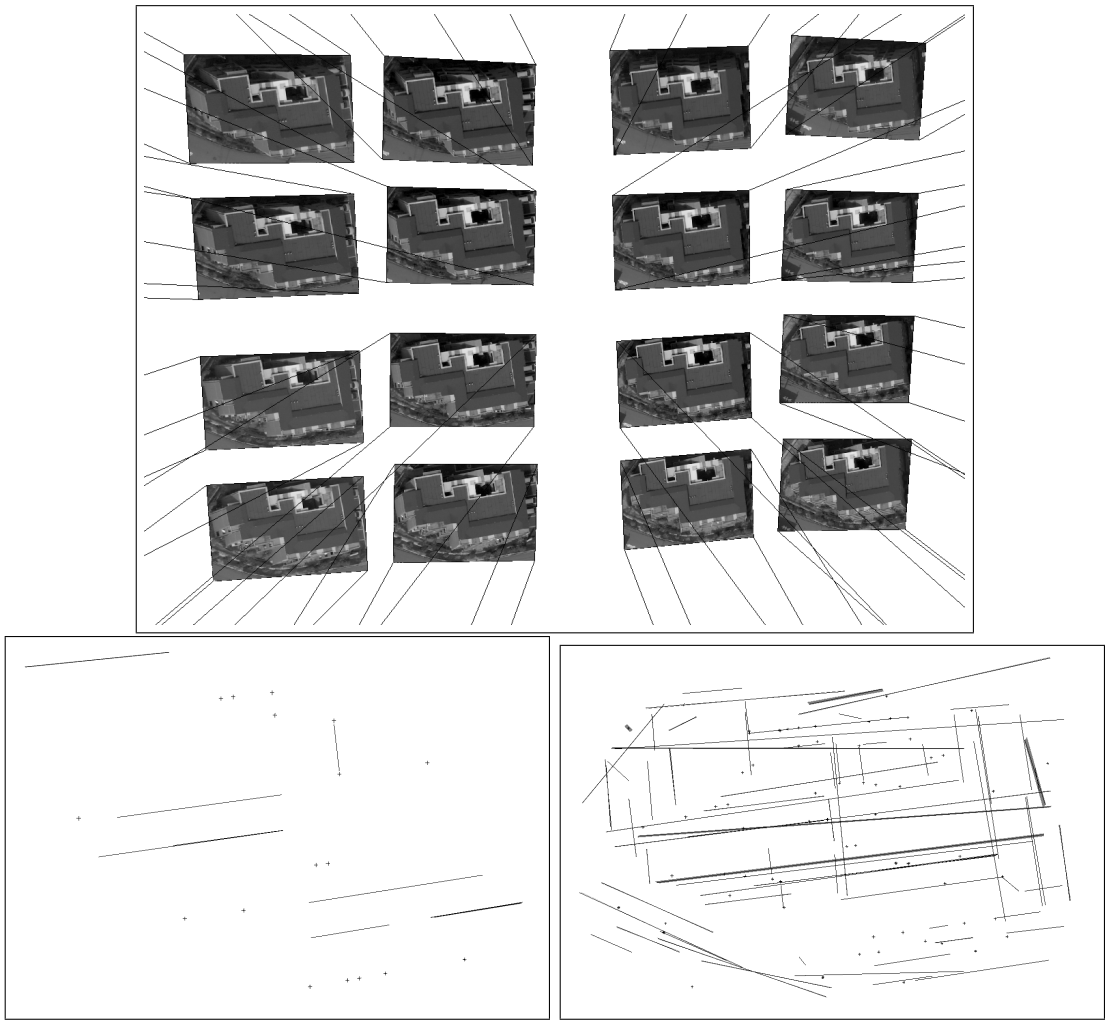

Figure 4.18: *Top:* Geometric 3d configuration of the 16 images taken form a large complex house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 17 3d points and 10 3d lines. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 71 3d points and 74 3d lines.

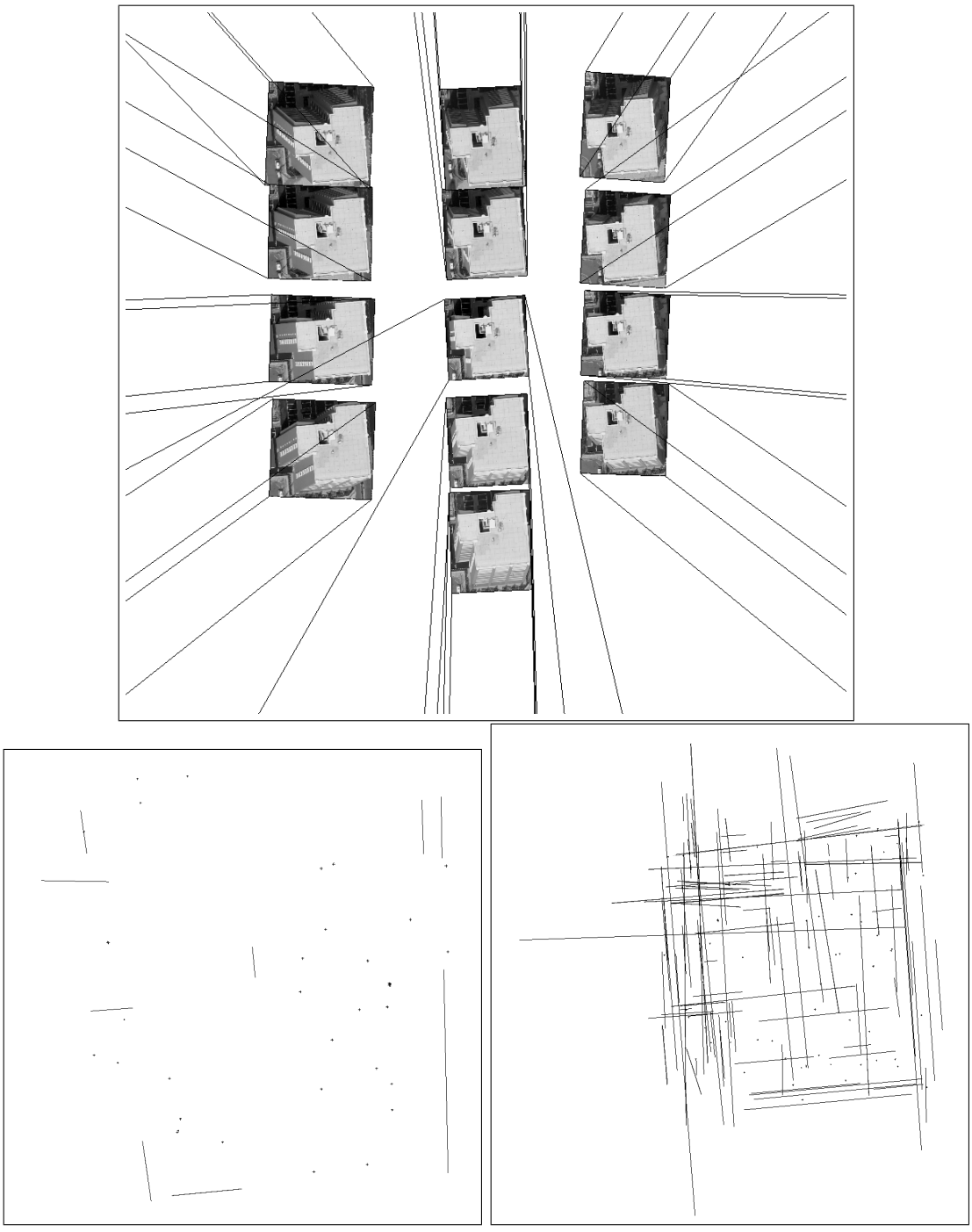

Figure 4.19: *Top:* Geometric 3d configuration of the 13 images taken form a large complex house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 35 3d points and 9 3d lines. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 66 3d points and 105 3d lines.

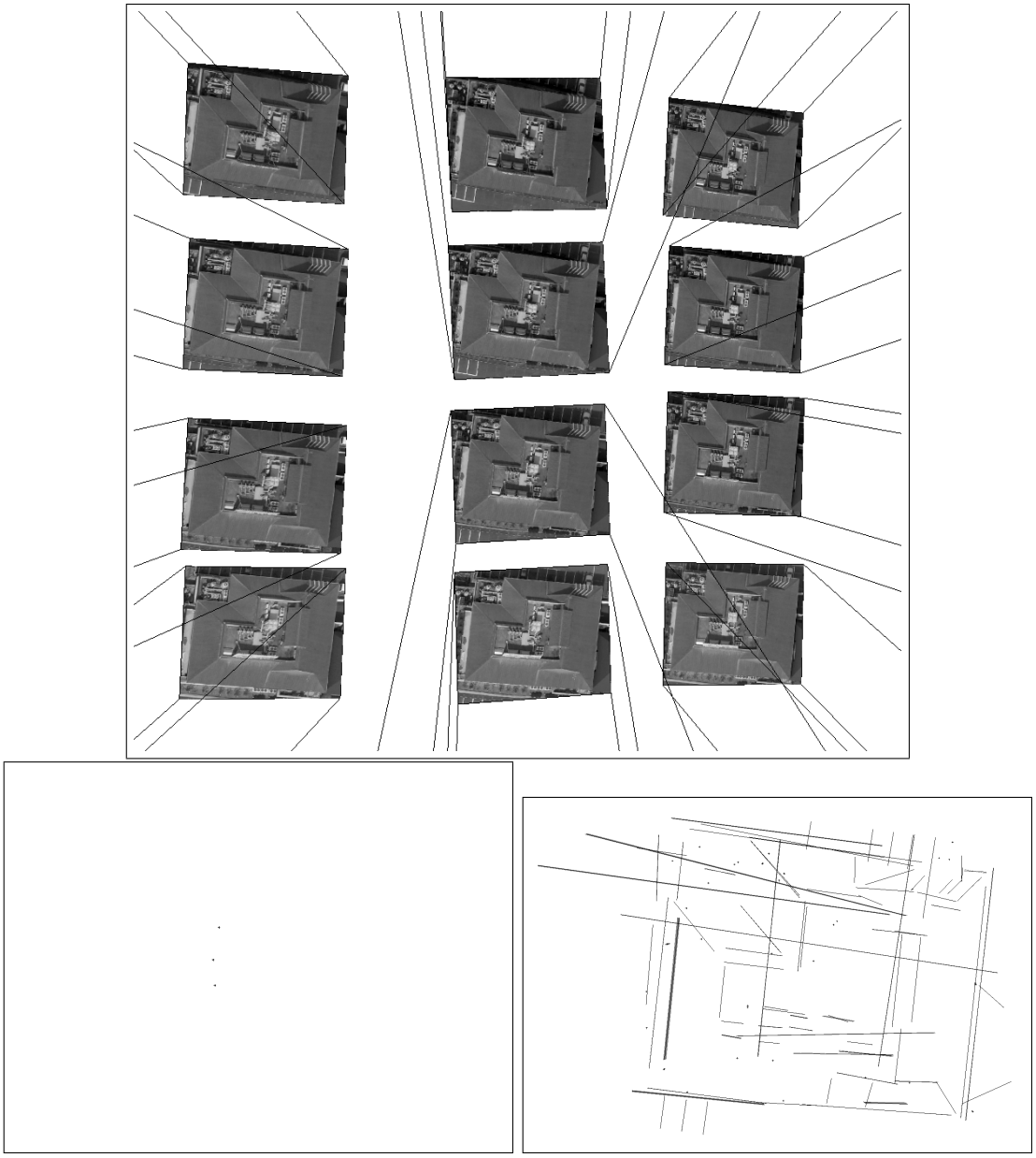

Figure 4.20: *Top:* Geometric 3d configuration of the 12 images taken form a large complex house. The lines indicate the connection of the image corners with the projection center. *Bottom left:* 3d reconstruction result from the clique partition grouping method. There are 3 3d points and no 3d lines. *Bottom right:* 3d reconstruction result from the entropy bound grouping method. There are 34 3d points and 80 3d lines.

Observe that the presented reconstructions are solely based on the image geometry. Apart from the feature extraction, the image content is not used for matching features at all. Nonetheless, reasonable results can be achieved without using any image content, strengthening the vital importance of geometry for multi image feature matching and fully automatic scene reconstruction. It has been shown that, in order to match features based solely on their geometric configuration, strong geometric evidence, i.e. many images, are required. Additional geometric cues resulting from the application, such as orthogonality, parallelity or symmetry assumptions, may also be included in order to obtain better results. However, this path has not been followed here.

As modern digital cameras produce many more images at very low extra expense, the importance of geometric grouping for the task of photogrammetric building reconstruction will become more important than ever in the future. Also the very improved noise levels of modern digital cameras make it possible, to work with fully automatic feature extraction that must be seen as one of the major bottlenecks for the presented reconstructions.

Chapter 5

Conclusion

A purely geometric framework for the 3d reconstruction of objects from multiple oriented and calibrated images has been presented. It has been shown that grouping geometric entities is an adequate approach for solving the 3d reconstruction problem on a geometric basis. The possibility of solving the matching problem without referring to the actual appearance of the objects in the images suggests that geometric cues are very strong by themselves and should not be neglected in any feature matching approach. It has been seen that highly redundant observations are required, if the image content is neglected. Though one should not neglect the image content in practical applications, the strong evidence contained in the geometry has become extremely cheap in recent years through the wide availability of digital cameras.

All results presented here were generated fully automatic without any human interaction including per image parameter tuning. Also note that no explicit high level building model knowledge, such as orthogonality, parallelity and symmetry or even building prototypes, has been used for generating the 3d reconstructions in the previous section. In my opinion the most promising approach for generating building models would be to fit explicit high level building models to the generated data. The sparseness of the data suggests that the computational burden is manageable, although this has not been investigated. Furthermore it could be helpful, to include the image content into the approach (for instance into the consistency requirement of the consistent, sufficient and irreducible subsets of observations presented in section 3.2), although I would suspect the performance gain to be minor, because of the weak radiometric information contained at scene corners and scene edges. Improving the performance of feature extraction, increasing the geometric redundancy by adding more images or including other sensors such as laser scanners seems much more promising. Note that due to the purely geometric nature of the approach, this sensor fusion is straightforward, as no image specific information is used.

The contribution of this work is the introduction of uncertain oriented projective geometry. By doing so, new compound entities such as line segments and polygons in 2d and 3d space as well as 2d edgels and 3d facets became representable in the framework of Heuel [2004]. The bi-linear relations occurring in this augmented framework enabled the design of a tree-like data structure for efficiently performing large amounts of statistical tests of relations between the uncertain oriented projective entities, which is the second major contribution of this work and made the application of the framework possible for large real world datasets.

The third contribution is the formulation of various computer vision tasks including the 3d reconstruction of points and lines as geometric grouping problems and devising strategies for solving those. The final major contribution of this work is the analysis of the entropy change incurred by incremental estimation in the Gauss-Helmert-model with constraints.

A unified information theoretic measure has been devised that balances efficiency against robustness for making incremental grouping decisions. The novel approach has been shown to perform well compared to the existing methods on a synthetic dataset with known ground truth. Also the results for automatic reconstruction of buildings from digital aerial images obtained with this information theoretic approach looked promising and yielded results for weaker configurations, where the graph based methods were not able to yield a solution.

There remain several interesting paths of research to follow. Fitting explicit high-level building prototypes to the very sparse sets of points and lines obtained by the geometric grouping seems a promising path to go for solving the problem of fully automatic building reconstruction.

Also the integration of other geometry sensors, such as laser scanners, is a very interesting topic. The uncertain 3d point clouds obtained by a laser scanner are straightforwardly integrated into the presented framework. That the redundancy of such 3d point clouds is extremely high is also very promising, although it must be studied, in how far other data structures might be required in order to cope with the huge amounts of data that do not fit into the fast memory any more.

From a practical point of view it would be very important to integrate other primitives, such as cylinders and cones into the framework. The research problems faced in this case are that the relations are no longer bi-linear but quadratic. A major problem faced in this case is that the presented data structure is no longer applicable in this case and one has to come up with an alternative for dealing with large real world datasets. However, the presented grouping strategies including the entropy bound method should also work in this case using first order approximations of the quadratic relations, which looks a very promising approach.

Finally it must be remarked that still the performance of low level feature extraction presents one of the major bottlenecks for the fully automatic feature based reconstruction from images. Hence, in order to improve the performance and robustness of the presented framework, increasing the robustness and accuracy of automatic feature extraction methods is another practical important topic for further research.

Bibliography

[Amir and Lindenbaum, 1998] A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):168–185, 1998.

[Ausiello *et al.*, 1999] G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag New York, Inc., 1999.

[Baillard and Zisserman, 1999] C. Baillard and A. Zisserman. Automatic reconstruction of piecewise planar models from multiple views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2559–2565, 1999.

[Baillard and Zisserman, 2000] C. Baillard and A. Zisserman. A plane-sweep strategy for the 3D reconstruction of buildings from multiple images. In *Proceedings of the 19th ISPRS Congress*, Amsterdam, July 2000.

[Baillard *et al.*, 1999] 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.

[Baltsavias *et al.*, 2001] E.P. Baltsavias, A. Grün, and L. VanGool, editors. *Automatic Extraction of Man-Made Objects from Aerial and Satellite Images III*. A.A.Balkema Publishers, Ascona, June 2001.

[Baltsavias, 2004] E. Baltsavias. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58:129–151, 2004.

[Bartoli and Sturm, 2003] Adrien Bartoli and Peter F. Sturm. Multiple-view structure and motion from line correspondences. In *Proceedings of the International Conference on Computer Vision*, pages 207–212, 2003.

[Bartoli and Sturm, 2005] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding*, 100(3):416–441, 2005.

[Bartoli, 2001] Adrien Bartoli. Piecewise planar segmentation for automatic scene modeling. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 283–289, 2001.

[Baumberg, 2000] Adam Baumberg. Reliable feature matching across widely separated views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1774–1781, 2000.

[Bay *et al.*, 2005] H. Bay, V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline stereo matching with line segments. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[Bayer and McCreight, 1972] R. Bayer and E. McCreight. Organization and maintenance of large ordered indexes. *Acta Informatica*, 1:173–189, 1972.

[Bazin and Vezien, 2005] Pierre-Louis Bazin and Jean-Marc Vezien. Integration of geometric elements, euclidean relations, and motion curves for parametric shape and motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1960–1976, 2005.

[Beardsley *et al.*, 1996] Paul A. Beardsley, Philip H. S. Torr, and Andrew Zisserman. 3d model acquisition from extended image sequences. In *Proceedings of European Conference on Computer Vision*, pages 683–695, 1996.

[Beckmann *et al.*, 1990] N. Beckmann, H. Kriegel, R. Schneider, and R.Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the ACM SIGMOD Symposium on Principles of Database Systems*, pages 322–331, 1990.

[Beder and Förstner, 2006a] Christian Beder and Wolfgang Förstner. Direct solutions for computing cylinders from minimal sets of 3d points. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Proceedings of the European Conference on Computer Vision*, number 3951 in LNCS, pages 135–146. Springer, 2006.

[Beder and Förstner, 2006b] Christian Beder and Wolfgang Förstner. Direkte Bestimmung von Zylindern aus 3D-Punkten ohne Nutzung von Oberflächennormalen. In Thomas Luhmann and Christina Müller, editors, *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik*, pages 206–213. Herbert Wichmann Verlag, 2006.

[Beder and Steffen, 2006] Christian Beder and Richard Steffen. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In K. Franke, K.-R. Müller, B. Nickolay, and R. Schäfer, editors, *Pattern Recognition*, number 4174 in LNCS, pages 657–666. Springer, 2006.

[Beder, 2004a] Christian Beder. Fast statistically geometric reasoning about uncertain line segments in 2d- and 3d-space. In C.E.Rasmussen, H.H.Bülthoff, B.Schölkopf, and M.A.Giese, editors, *Pattern Recognition*, number 3175 in LNCS, pages 375–382. Springer, 2004.

[Beder, 2004b] Christian Beder. A unified framework for the automatic matching of points and lines in multiple oriented images. In *Proceedings of the 20th ISPRS Congress, Istanbul, Turkey*, pages 1109–1113. ISPRS, 2004.

[Beder, 2005] Christian Beder. Agglomerative grouping of observations by bounding entropy variation. In Walter Kropatsch, Robert Sablatnig, and Allan Hanbury, editors, *Pattern Recognition*, number 3663 in LNCS, pages 101–108. Springer, 2005.

[Begelfor and Werman, 2005] Evgeni Begelfor and Michael Werman. How to put probabilities on homographies. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1666–1670, 2005.

[Bignone *et al.*, 1996] Frank Bignone, Olof Henricsson, Pascal Fua, and Markus A. Stricker. Automatic extraction of generic house roofs from high resolution aerial imagery. In *Proceedings of European Conference on Computer Vision*, pages 85–96, 1996.

[Blaschke, 1948] Wilhelm Blaschke. *Projektive Geometrie*. Wolfenbütteler Verlagsanstalt K.G., 1948.

[Boyer and Sarkar, 1999] Kim L. Boyer and Sudeep Sarkar. Perceptual organization in computer vision: status, challenges, and potential. *Comput. Vis. Image Underst.*, 76(1):1–5, 1999.

[Brand, 1999] M. Brand. Pattern discovery via entropy minimization. In *Proceedings of Seventh International Workshop on Artificial Intelligence and Statistics,* Fort Lauderdale, Florida. Morgan Kaufmann, January 1999.

[Braun *et al.*, 1994] C. Braun, T. H. Kolbe, F. Lang, W. Schickler, V. Steinhage, A. B. Cremers, W. Förstner, and L. Plümer. Models for photogrammetric building reconstruction. *Computer & Graphics*, 19(1):109–118, 1994.

[Bretar and Roux, 2005] Frederic Bretar and Michel Roux. Hybrid image segmentation using lidar 3d planar primitives. In *Proc. of ISPRS WG III/3,III/4,V/3 Workshop Laser scanning 2005*, pages 72–78, 2005.

[Bronstein *et al.*, 2001] I.N. Bronstein, K.A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Harri Deutsch, 2001.

[Brown and Lowe, 2002] Matthew Brown and David Lowe. Invariant features from interest point groups. In *Proceedings of the British Machine Vision Conference*, 2002.

[Canny, 1986] John F. Canny. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[Carneiro and Jepson, 2002] G. Carneiro and Allan D. Jepson. Phase-based local features. In *Proceedings of European Conference on Computer Vision*, pages 282–296, London, UK, 2002. Springer-Verlag.

[Chater and Vitanyi, 2003a] Nick Chater and Paul M.B. Vitanyi. The generalized universal law of generalization. *Journal of Mathematical Psychology*, 47(3):346–369, 2003.

[Chater and Vitanyi, 2003b] Nick Chater and Paul M.B. Vitanyi. Simplicity: A unifying principle in cognitive science? *Trends in Cognitive Sciences*, 7(1):19–22, 2003.

[Cheeseman, 1985] Peter Cheeseman. In defense of probability. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 1002–1009, Los Angeles, California, 1985. Morgan Kaufmann.

[Cheng *et al.*, 2001] Y.-Q. Cheng, X.G. Wang, R.T. Collins, E.M. Riseman, and A.R. Hanson. Three-dimensional reconstruction of points and lines with unknown correspondence across images. *International Journal of Computer Vision*, 45(2):129 – 156, 2001.

[Chiba and Kanade, 1998] N. Chiba and Takeo Kanade. A tracker for broken and closely spaced lines. In *Proceedings of the 1996 International Society for Photogrammetry and Remote Sensing Conference (ISPRS '98)*, pages 676 – 683, 1998.

[Clarke *et al.*, 1996] J. C. Clarke, S. Carlsson, and Andrew Zisserman. Detecting and tracking linear features efficiently. In *Proceedings of the British Machine Vision Conference*, 1996.

[Collins *et al.*, 1998] R.T. Collins, C.O. Jaynes, Y.-Q. Cheng, X. Wang, F. Stolle, E.M. Riseman, and A.R. Hanson. The ascender system: Automated site modeling from multiple aerial images. *Computer Vision and Image Understanding*, 72(2):143–162, 1998.

[Cormen *et al.*, 1990] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Intorduction to Algorithms*. MIT Press, 1990.

[Cover and Thomas, 1991] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[Crevier, 1999] Daniel Crevier. A probabilistic method for extracting chains of collinear segments. *Computer Vision and Image Understanding*, 76(1):36–53, 1999.

[Criminisi *et al.*, 1999] A. Criminisi, I. Reid, and A. Zisserman. A plane measuring device. *Image and Vision Computing*, 17(8):625–634, 1999.

[Csurka *et al.*, 1997] Gabriella Csurka, Cyril Zeller, Zhengyou Zhang, and Olivier Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer Vision and Image Understanding*, 68(1):18–36, 1997.

[Davison and Murray, 2002] A. J. Davison and D. W. Murray. Simultaneous localisation and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, July 2002.

[Davison, 2003] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision*, pages 1403–1410, October 2003.

[Davison, 2005] Andrew J. Davison. Active search for real-time vision. In *Proceedings of the International Conference on Computer Vision*, pages 66–73, 2005.

[Deriche and Giraudon, 1993] Rachid Deriche and Gerard Giraudon. A computational approach for corner and vertex detection. *International Journal on Computer Vision*, 10(2):101–124, 1993.

[DeVore, 2005] Michael D. DeVore. Estimates of error probability for complex gaussian channels with generalized likelihood ratio detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1580–1591, 2005.

[Dick *et al.*, 2000] Anthony R. Dick, Philip H. S. Torr, and Roberto Cipolla. Automatic 3d modelling of architecture. In *Proceedings of the British Machine Vision Conference*, 2000.

[Dick *et al.*, 2001] Anthony R. Dick, Philip H. S. Torr, Simon J. Ruffle, and Roberto Cipolla. Combining single view recognition and multiple view stereo for architectural scenes. In *Proceedings of the International Conference on Computer Vision*, pages 268–274, 2001.

[Dick *et al.*, 2002] Anthony R. Dick, Philip H. S. Torr, and Roberto Cipolla. A bayesian estimation of building shape using mcmc. In *Proceedings of European Conference on Computer Vision*, pages 852–866, 2002.

[Dick *et al.*, 2004] A. Dick, P. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2):111 – 134, 2004.

[Duda *et al.*, 2001] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, 2001.

[Estrada and Jepson, 2004] Francisco J. Estrada and Allan D. Jepson. Perceptual grouping for contour extraction. In *ICPR (2)*, pages 32–35, 2004.

[Faugeras and Luong, 2001] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images - The Laws that Govern the Formation of Mutliple Images of a Scene and some of their Applications*. MIT Press, 2001.

[Faugeras, 1993] Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[Feller, 1968] William Feller. *An Introduction to Probability Theory and Its Applications*. Wiley, 1968.

[Finkel and Bentley, 1974] R. Finkel and J. Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974.

[Fischer *et al.*, 1998] A. Fischer, T. H. Kolbe, F. Lang, A. B. Cremers, W. Förstner, L. Plümer, and V. Steinhage. Extracting buildings from aerial images using hierarchical aggregation in 2d and 3d. *Computer Vision and Image Understanding*, 72(2):185–203, 1998.

[Fischler and Bolles, 1981] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[Förstner and Gülch, 1987] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *ISPRS Intercommission Workshop, Interlaken*, 1987.

[Förstner and Plümer, 1997] W. Förstner and L. Plümer, editors. *Semantic Modeling for the Acquisition of Topographic Information from Images and maps*. Birkhäuser, 1997.

[Förstner and Wrobel, 2004] Wolfgang Förstner and Bernhard Wrobel. Mathematical concepts in photogrammetry. In J.C.McGlone, E.M.Mikhail, and J.Bethel, editors, *Manual of Photogrammetry*, pages 15–180. ASPRS, 2004.

[Förstner *et al.*, 2000] W. Förstner, A. Brunn., and S. Heuel. Statistically testing uncertain geometric relations. In *Proceedings of the DAGM, Kiel, Germany*, pages 17–26, 2000.

[Förstner, 1983] W. Förstner. Reliablitiy and discernability of extended gauss-markov models. In F. Ackermann, editor, *Seminar Mathematical Models of Geodetic/Photogrammetric Point Determiation with regard to Outliers and Systematic Errors*, pages 79–103, München, 1983. Deutsche Geodätische Kommission, A98.

[Förstner, 1987] Wolfgang Förstner. Reliability Analysis of Parameter Estimation in Linear Models with Applications to Mensuration Problems in Computer Vision. *Computer Vision, Graphics, and Image Processing*, 40:273–310, 1987.

[Förstner, 1990] Wolfgang Förstner. Modelle intelligenter Bildsensoren und ihre Qualität. In R. E. Großkopf, editor, *Mustererkennung 1990*, pages 1–21. Springer, 1990.

[Förstner, 1994] W. Förstner. A framework for low level feature extraction. In *European Conference on Computer Vision*, pages 383–394, 1994.

[Förstner, 1998] Wolfgang Förstner. Image preprocessing for feature extraction in digital intensity, color and range images. In *Proceedings of the International Summer School on* Data Analysis and the Statistical Foundations of Geomatics*, Chania, Crete*, 1998.

[Förstner, 1999] W. Förstner. 3d-city models: Automatic and semiautomatic acquisition methods. In *Photogrammetrische Woche*, Stuttgart, September 1999.

[Förstner, 2001a] W. Förstner. Algebraic projective geometry and direct optimal estimation of geometric entities. In *Proceedings of the OeAGM*, 2001.

[Förstner, 2001b] W. Förstner. Generic estimation procedures for orientation with minimum and redundant information. In A. Gruen and T. S. Huang, editors, *Calibration and Orientation of Cameras in Computer Vision*, number 34 in Series in Information Sciences. Springer, 2001.

[Förstner, 2001c] 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.

[Förstner, 1989] W. Förstner. Image analysis techniques for digital photogrammetry. *Schriftenreihe des Instituts für Photogrammetrie der Universität Stuttgart*, 13:205–221, 1989.

[Früh and Zakhor, 2003] Christian Früh and Avideh Zakhor. Constructing 3d city models by merging ground-based and airborne views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 562–569, 2003.

[Fuchs, 1998] C. Fuchs. *Extraktion polymorpher Bildstrukturen und ihre topologische und geometrische Gruppierung*. DGK, Bayer. Akademie der Wissenschaften, München, Reihe C, Heft 502, 1998.

[Garey, 1979] Johnson. D. S. Garey, M.R. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[Georgescu and Meer, 2004] Bogodan Georgescu and Peter Meer. Point matching under large image deformations and illumination changes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(6):674–688, 2004.

[Goldberger, 2005] Jacob Goldberger. Reconstructing camera projection matrices from multiple pairwise overlapping views. *Computer Vision and Image Understanding*, 97(3):283–296, 2005.

[Goldschmidt *et al.*, 1996] Oliver Goldschmidt, Dorit S. Hochbaum, Cor Hurkens, and Gang Yu. Approximation algorithms for the k-clique covering problem. *SIAM Journal on Discretete Mathematics*, 9(3):492–509, 1996.

[Gool *et al.*, 2002] L. Van Gool, T. Tuytelaars, V. Ferrari, C. Strecha, J. Vanden Wyngaerd, and M. Vergauwen. 3d modeling and registration under wide baseline conditions. In *Proceedings of the ISPRS Symposium on Photogrammetric Computer Vision*, pages 3–14, 2002.

[Gouet *et al.*, 1998] V. Gouet, P. Montesinos, and D. Pelé. A fast matching method for color uncalibrated images using differential invariants. In *Proceedings of the British Machine Vision Conference*, volume I, pages 367–376, Southampton, UK, September 1998.

[Grünwald and Dawid, 2004] Peter Grünwald and A Philip Dawid. Game theory, maximum entropy, minimum discrepancy, and robust bayesian decision theory. *Annals of Statistics*, 32(4):1367–1433, 2004.

[Gruber *et al.*, 2003] M. Gruber, F. Leberl, and R. Perko. Paradigmenwechsel in der photogrammetrie durch digitale luftbildaufnahme? *Photogrammetrie, Fernerkundung und Geoinformation*, 4:285–297, 2003.

[Grün *et al.*, 1995] A. Grün, O. Kübler, and P. Agouris, editors. *Automatic Extraction of Man-Made Objects from Aerial and Space Images*. Birkhäuser, 1995.

[Grün *et al.*, 1997] A. Grün, E.P. Baltsavias, and O. Henricsson, editors. *Automatic Extraction of Man-Made Objects from Aerial and Space Images II*. Birkhäuser, 1997.

[Grünwald and Halpern, 2004] Peter D. Grünwald and Joseph Y. Halpern. When ignorance is bliss. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 226–234, Arlington, Virginia, United States, 2004. AUAI Press.

[Grünwald and Vitanyi, 2003] Peter D. Grünwald and Paul M. B. Vitanyi. Kolmogorov complexity and information theory. with an interpretation in terms of questions and answers. *J. of Logic, Lang. and Inf.*, 12(4):497–529, 2003.

[Grünwald and Vitányi, 2004] Peter Grünwald and Paul M. B. Vitányi. Shannon information and kolmogorov complexity. *CoRR*, cs.IT/0410002, 2004.

[Guttman, 1984] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47–57, 1984.

[Guy and Medioni, 1996] Gideon Guy and Gérard Medioni. Inferring global perceptual contours from local features. *International Journal on Computer Vision*, 20(1-2):113–133, 1996.

[Guy and Medioni, 1997] Gideon Guy and Gérard G. Medioni. Inference of surfaces, 3d curves, and junctions from sparse, noisy, 3d data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1265–1277, 1997.

[Harris and Stephens, 1988] C.G. Harris and M.J. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, 1988.

[Hartley and Zisserman, 2000] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[Hartley, 1998] Richard I. Hartley. Chirality. *International Journal on Computer Vision*, 26(1):41–61, 1998.

[Heitger, 1995] F. Heitger. Feature detection using suppression and enhancement. Technical Report TR-163, Image Science Lab, ETH Zürich, Switzerland, 1995.

[Helava, 1976] U. V. Helava. Digital correlation in photogrammric instruments. In *International Archives for Photogrammetry, Vol. 23, Part II*, 1976.

[Henricsson, 1997] Olof Henricsson. 3-d building reconstruction with aruba: A qualitative and quantitative evaluation. In A. Gruen, E.P. Baltsavias, and O. Henricsson, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, page 65. Birkhäuser Verlag, Basel, Switzerland, 1997.

[Henricsson, 1998] Olof Henricsson. The role of color attributes and similarity grouping in 3-d building reconstruction. *Computer Vision and Image Understanding*, 72(2):163–184, 1998.

[Heuel and Förstner, 2001] S. Heuel and W. Förstner. Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 2001.

[Heuel, 2001] 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.

[Heuel, 2004] Stephan Heuel. *Uncertain Projective Geometry - Statistical Reasoning for Polyhedral Object Reconstruction*. Springer, 2004.

[Heyden and Pollefeys, 2004] Anders Heyden and Marc Pollefeys. Multiple view geometry. In G. Medioni and S.B. Kang, editors, *Emerging Topics in Computer Vision*, pages 45–107. Prentice-Hall, 2004.

[Hofmann and Buhmann, 1997] Thomas Hofmann and Joachim M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.

[Horn, 1986] B.K.P. Horn. *Robot Vision*. MIT Press, 1986.

[Horn, 1990] B. K. P. Horn. Relative orientation. *IJCV*, 4(1):59–78, 1990.

[Huber, 1981] P. J. Huber. *Robust Statistics*. Wiley New York, 1981.

[Illingworth and Kittler, 1988] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision and Graphical Image Processing*, 44(1):87–116, 1988.

[Jaynes *et al.*, 1997] Christopher O. Jaynes, Allen Hanson, and Edward Riseman. Model-based surface recovery of buildings in optical and ranges images. In *Semantic Modeling for the Acquisition of Topografic Information from Images and Maps*, pages 211–227, 1997.

[Jaynes, 1982] E.T. Jaynes. On the rationale of maximum entropy methods. *Proceedings of the IEEE*, 70:939–952, 1982.

[Jung and Paparoditis, 2003] 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 *Proceedings of the ISPRS Conference on Photogrammetric Image Analysis*, pages 39–44, 2003.

[Kadir and Brady, 2001] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal on Computer Vision*, 45(2):83–105, 2001.

[Kadir *et al.*, 2004] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*. Springer, may 2004.

[Kagan *et al.*, 1973] A. M. Kagan, J. V. Linnik, and C. R. Rao. *Characterization problems in mathematical statistics*. Wiley, New York, 1973.

[Kahl, 2005] Fredrik Kahl. Multiple view geometry and the l_∞-norm. In *Proceedings of the International Conference on Computer Vision*, pages 1002–1009, 2005.

[Kanatani and Morris, 2001] Kenichi Kanatani and Daniel D. Morris. Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy. *IEEE Transactions on Information Theory*, 47(5):2017–2028, July 2001.

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

[Kanatani, 2004] Kenichi Kanatani. Uncertainty modeling and model selection for geometric inference. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(10):1307–1319, 2004.

[Kemp and Drummond, 2005] Christopher Kemp and Tom Drummond. Dynamic measurement clustering to aid real time tracking. In *Proceedings of the International Conference on Computer Vision*, pages 1500–1507, 2005.

[Khuller and Raghavachari, 1996] Samir Khuller and Balaji Raghavachari. Improved approximation algorithms for uniform connectivity problems. *J. Algorithms*, 21(2):434–450, 1996.

[Khuller, 1997] Samir Khuller. Approximation algorithms for finding highly connected subgraphs. In D.Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 236–265. PWS Publishing Company, 1997.

[Kim and Nevatia, 1999] ZuWhan Kim and Ramakant Nevatia. Uncertain reasoning and learning for feature grouping. *Comput. Vis. Image Underst.*, 76(3):278–288, 1999.

[Kim and Nevatia, 2004] Z.W. Kim and R. Nevatia. Automatic description of complex buildings from multiple images. *Computer Vision and Image Understanding*, 96(1):60–95, 2004.

[Kim *et al.*, 2001a] Z. Kim, A. Huertas, and R. Nevatia. A model-based approach for multiview complex building description. In *Proceedings of ASCONA*, 2001.

[Kim *et al.*, 2001b] Zu Whan Kim, Andres Huertas, and Ramakant Nevatia. Automatic description of buildings with complex rooftops from multiple images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 272–, 2001.

[Kim *et al.*, 2002] Z. Kim, A. Huertas, and R. Nevatia. Automatic description of complex buildings with multiple images. In *WACV*, pages 155–162, 2002.

[Knuth, 1998a] D. E. Knuth. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming*. Addison-Wesley, 1998.

[Knuth, 1998b] D. E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, 1998.

[Koch and Frahm, 2001] Reinhard Koch and Jan-Michael Frahm. Visual-geometric scene reconstruction from image streams. In *6th International Workshop on Vision, Modeling and Visualization*, 2001.

[Koch *et al.*, 1998a] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Automatic 3d model acquisition from uncalibrated image sequences. In *Proceedings of Computer Graphics International*, pages 597–604, 1998.

[Koch *et al.*, 1998b] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proceedings of the European Conference on Computer Vision*, 1998.

[Koch *et al.*, 1999a] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Realistic 3-d scene modeling from uncalibrated image sequences. In *Proceedings of the International Conference for Image Processing*, 1999.

[Koch *et al.*, 1999b] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Robust calibration and 3d geometric modeling from large collections of uncalibrated images. In W. Förstner, J. Buhmann, A. Faber, and P. Faber, editors, *Proceedings of the DAGM*, Informatik Aktuell, pages 412–420. Springer, 1999.

[Koch *et al.*, 2000] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Realistic surface reconstruction of 3d scenes from uncalibrated image sequences. *Journal of Visualization and Computer Animation*, 11:115–127, 2000.

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

[Koch, 1993] Reinhard Koch. Automatic reconstruction of buildings from stereoscopic image sequences. In *Proceedings of the Eurographics*, 1993.

[Koch, 1994a] Reinhard Koch. 3-d scene modeling from stereoscopic image sequences. In Y. Paker and S. Wilbur, editors, *Image Processing for Broadcast and Video Production*, pages 128–135. Springer, 1994.

[Koch, 1994b] Reinhard Koch. Model-based 3-d scene analysis from stereoscopic image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 49:23–30, 1994.

[Koch, 1995] Reinhard Koch. 3-d surface reconstruction from stereoscopic image sequences. In *Proceedings of the International Conference of Computer Vision*, 1995.

[Koch, 1996] Reinhard Koch. Surface segmentation and modeling of 3-d polygonal objects from stereoscopic image pairs. In *Proceedings of the International Conference on Pattern Recognition*, 1996.

[Koch, 1997] Karl-Rudolf Koch. *Parameterschätzung und Hypothesentests*. Dümmler, 1997.

[Koenderink and van Doorn, 1987] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, 1987.

[Kontkanen *et al.*, 2004] Petri Kontkanen, Petri Myllymäki, Wray Buntine, Jorma Rissanen, and Henry Tirri. An mdl framework for data clustering. In P.M.B. Vitanyi, editor, *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2004.

[Kosecka and Zhang, 2005] Jana Kosecka and Wei Zhang. Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding*, 100(3):274–293, 2005.

[Kröpfl *et al.*, 2004] M. Kröpfl, E. Kruck, and M. Gruber. Geometric calibration of the digital large format aerial camera ultracam-d. In *Proceedings of the 20th ISPRS Congress, Istanbul, Turkey*. ISPRS, 2004.

[Lang and Förstner, 1996] F. Lang and W. Förstner. Surface reconstruction of man-made objects using polymorphic mid-level features and generic scene knowledge. In *18th ISPRS Congress, Wien, Proceedings*, 1996.

[Liedtke and Koch, 1993] C.-E. Liedtke and R. Koch. Modelling of 3d scenes from the analysis of stereoscopic image sequences. *International Journal on Pattern Recognition and Image Analysis*, 3:346–351, 1993.

[Lin and Nevatia, 1995] C. Lin and R. Nevatia. 3-d descriptions of buildings from an oblique view aerial image. In *IEEE International Symposium on Computer Vision*, pages 377–382, 1995.

[Lin and Nevatia, 1996] C. Lin and R. Nevatia. Buildings detection and description from monocular aerial images. In *ARPA Image Understanding Workshop*, 1996.

[Lin and Nevatia, 1998] Chungan Lin and Ramakant Nevatia. Building detection and description from a single intensity image. *Computer Vision and Image Understanding*, 72(2):101–121, 1998.

[Lindeberg, 1998a] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal on Computer Vision*, 30(2):117–156, 1998.

[Lindeberg, 1998b] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal on Computer Vision*, 30(2):79–116, 1998.

[Lowe, 1985] David G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.

[Lowe, 1999] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.

[Lowe, 2004] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110, 2004.

[Lucas and Kanade, 1981] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.

[Luxen and Förstner, 2001] M. Luxen and W. Förstner. Optimal Camera Orientation from Points and Straight Lines. In B. Radig and S. Florczyk, editors, *Pattern Recognition, 23rd DAGM Symposium*, number 2191 in LNCS, pages 84–91. Springer, September 2001.

[Marr, 1982] David Marr. *Vision*. W. H. Freeman and Company, New York, 1982.

[Matas and Chum, 2005] Jiri Matas and Ondrej Chum. Randomized ransac with sequential probability ratio test. In *Proceedings of the International Conference on Computer Vision*, pages 1727–1732, 2005.

[Matas *et al.*, 2002] J. Matas, O.Chum, M.Urban, and T.Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002.

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

[Mayer, 2004] Helmut Mayer. Object extraction for digital photogrammetric workstations. In *Proceedings of the ISPRS*, page 414, 2004.

[McCafferty, 1990] J.D. McCafferty. *Human And Machine Vision*. Ellis Horwood, 1990.

[Medioni *et al.*, 2000] Gerard Medioni, Chi-Keung Tang, and Mi-Suen Lee. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.

[Meltzer *et al.*, 2004a] Jason Meltzer, Rakesh Gupta, Ming-Hsuan Yang, and Stefano Soatto. Simultaneous localization and mapping using multiple view feature descriptors. In *Proceedings of the IROS*, 2004.

[Meltzer *et al.*, 2004b] Jason Meltzer, Ming-Hsuan Yang, Rakesh Gupta, and Stefano Soatto. Multiple view feature descriptors from image sequences via kernel principal component analysis. In *Proceedings of European Conference on Computer Vision*, pages 215–227, 2004.

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

[Mikolajczyk and Schmid, 2001] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pages 525–531, 2001.

[Mikolajczyk and Schmid, 2002] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142. Springer, 2002. Copenhagen.

[Mikolajczyk and Schmid, 2003] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 257–263, 2003.

[Mikolajczyk and Schmid, 2005] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[Mikolajczyk *et al.*, 2003] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *Proceedings of the British Machine Vision Conference*, 2003.

[Mikolajczyk *et al.*, 2005] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 2005. to appear.

[Mitchell, 1997] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[Mohan and Nevatia, 1992] Rakesh Mohan and Ramakant Nevatia. Perceptual organization for scene segmentation and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6):616–635, 1992.

[Molton *et al.*, 2004] N. D. Molton, A. J. Davison, and I. D. Reid. Locally planar patch features for real-time structure from motion. In *Proc. British Machine Vision Conference*. BMVC, September 2004.

[Moons *et al.*, 1998] Theo Moons, David Frère, Jan Vandekerckhove, and Luc J. Van Gool. Automatic modeling and 3d reconstruction of urban house roofs from high resolution aerial imagery. In *Proceedings of European Conference on Computer Vision*, pages 410–425, 1998.

[Moreels and Perona, 2005] Pierre Moreels and Pietro Perona. Evaluation of features detectors and descriptors based on 3d objects. In *Proceedings of the International Conference on Computer Vision*, pages 800–807, 2005.

[Morris *et al.*, 1999] Daniel D. Morris, Kenichi Kanatani, and Takeo Kanade. Uncertainty modeling for optimal structure from motion. In *Vision Algorithms Theory and Practice*. Springer LNCS, September 1999.

[Mugnier *et al.*, 2004] Clifford J. Mugnier, Wolfgang Förstner, Bernhard Wrobel, Fidel Paderes, and Riadh Munjy. The mathematics of photogrammetry. In J.C.McGlone, E.M.Mikhail, and J.Bethel, editors, *Manual of Photogrammetry*, pages 181–316. ASPRS, 2004.

[Mumford, 2000] David Mumford. The dawning of the age of stochasticity. In *Mathematics: frontiers and perspectives*, pages 197–218. Amer. Math. Soc., Providence, RI, 2000.

[Nevatia and Chung, 1992] R. Nevatia and C.K.R. Chung. Recovering building structures from stereo. In *IEEE WACV*, pages 64–73, 1992.

[Niemeier, 2001] Wolfgang Niemeier. *Augleichungsrechnung*. deGruyter-Verlag, 2001.

[Nistér, 2003] David Nistér. An efficient solution to the five-point relative pose problem. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 195–202, 2003.

[Nistér, 2004] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–777, 2004.

[Noronha and Nevatia, 1997] Sanjay Noronha and Ramakant Nevatia. Detection and description of buildings from multiple aerial images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 588–594, 1997.

[Noronha and Nevatia, 2001] Sanjay Noronha and Ramakant Nevatia. Detection and modeling of buildings from multiple aerial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):501–518, 2001.

[Ommer and Buhmann, 2003] Björn Ommer and Joachim M. Buhmann. A compositionality architecture for perceptual feature grouping. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, number 2683 in LNCS, pages 275–290. Springer, 2003.

[Pajdla *et al.*, 1998] Tomas Pajdla, Tomas Werner, and Vaclav Hlavac. Oriented projective reconstruction. In *Proceedings of the OeAGM*, 1998.

[Paparoditis *et al.*, 1998] N. Paparoditis, M. Cord, M. Jordan, and J.P. Cocquerez. Building detection and reconstruction from mid- and high-resolution aerial imagery. *Computer Vision and Image Understanding*, 72(2):122–142, 1998.

[Paz and Moran, 1981] A. Paz and S. Moran. Non-deterministic polynomial optimization problems and their approximations. *Theoretical Computer Science*, 15:251–277, 1981.

[Perwass *et al.*, 2005] C. Perwass, C. Gebken, and G. Sommer. Estimation of geometric entities and operators from uncertain data. In *27. Symposium für Mustererkennung, DAGM 2005, Wien, 29.8.-2.9.005*, LNCS, pages 459–467. Springer-Verlag, Berlin, Heidelberg, 2005.

[Philip, 1996] J. Philip. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *Photogrammetric Record*, 15(88):589–599, 1996.

[Philip, 1998] J. Philip. Critical point configurations of the 5-, 6-, 7- and 8-point algorithms for relative orientation. Technical report, Department of Mathematics, Royal Institute of Technology, Stockholm, 1998.

[Pollefeys and Gool, 1997a] Marc Pollefeys and Luc Van Gool. Self-calibration from the absolute conic on the plane at infinity. In *Proceedings of the CAIP*, pages 175–182, 1997.

[Pollefeys and Gool, 1997b] Marc Pollefeys and Luc Van Gool. A stratified approach to metric self-calibration. Technical report, KU Leuven, Belgium, 1997.

[Pollefeys and Gool, 1997c] Marc Pollefeys and Luc Van Gool. A stratified approach to metric self-calibration. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 407–412, 1997.

[Pollefeys and Gool, 1999] Marc Pollefeys and Luc Van Gool. Stratified self-calibration with the modulus constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):707–724, 1999.

[Pollefeys and Gool, 2000a] Marc Pollefeys and Luc Van Gool. Some geometric insight in self-calibration and critical motion sequences. Technical report, K.U. Leuven, Belgium, 2000.

[Pollefeys and Gool, 2000b] Marc Pollefeys and Luc Van Gool. Some issues on self-calibration and critical motion sequences. In *Proceedings of the Asian Conference on Computer Vision*, pages 893–898, 2000.

[Pollefeys and Gool, 2002] Marc Pollefeys and Luc Van Gool. From images to 3d models. *Communications of the ACM*, 45:50–55, 2002.

[Pollefeys *et al.*, 1996a] Marc Pollefeys, Luc Van Gool, and Andre Oosterlinck. The modulus constraint: a new constraint for self-calibration. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 349–353, 1996.

[Pollefeys *et al.*, 1996b] Marc Pollefeys, Luc Van Gool, and Andre Oosterlinck. Self-calibration with the modulus constraint. Technical report, KU Leuven, Belgium, 1996.

[Pollefeys *et al.*, 1997a] Marc Pollefeys, Luc Van Gool, and Andre Oosterlinck. Euclidean self-calibration via the modulus constraint. In F. Dillen, L. Vrancken, L. Verstraelen, and I. Van de Woestijne, editors, *Geometry and topology of submanifolds*, pages 283–291. VIII" World Scientific, 1997.

[Pollefeys *et al.*, 1997b] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction in spite of varying and unknwon internal camera parameters. Technical report, KU Leuven, Belgium, 1997.

[Pollefeys *et al.*, 1998a] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings of the International Conference on Computer Vision*, pages 90–95, 1998.

[Pollefeys *et al.*, 1998b] Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, and Luc Van Gool. Automatic generation of 3d models from photographs. In *Proceedings of Virtual Systems and MultiMedia*, 1998.

[Pollefeys *et al.*, 1998c] Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, and Luc Van Gool. Flexible 3d acquisition with a monocular camera. In *Proceedings of the International Conference on Robotics and Automation*, volume 4, pages 2771–2776, 1998.

[Pollefeys *et al.*, 1998d] Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, and Luc Van Gool. Flexible acquisition of 3d structure from motion. In *Proceedings of the IEEE workshop on Image and Multidimensional Digital Signal Processing*, pages 195–198, 1998.

[Pollefeys *et al.*, 1998e] Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, and Luc Van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. In *Proceedings of the SMILE Workshop*, number 1506 in LNCS, pages 138–153. Springer, 1998.

[Pollefeys *et al.*, 1999a] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. An automatic method for acquiring 3d models from photographs: applications to an archaeological site. In *Proceedings of the ISPRS International Workshop on Photogrammetric Measurement, Object Modeling and Documentation in Architecture and Industry*, pages 76–80, 1999.

[Pollefeys *et al.*, 1999b] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Hand-held acquisition of 3d models with a video camera. In *Proceedings of the IEEE Second International Conference on 3-D Digital Imaging and Modeling*, pages 14–23, 1999.

[Pollefeys *et al.*, 1999c] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–32, 1999.

[Pollefeys *et al.*, 2000a] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Automated reconstruction of 3d scenes from sequences of images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(4):251–267, 2000.

[Pollefeys *et al.*, 2000b] M. Pollefeys, M. Proesmans, R. Koch, M. Vergauwen, and L. Van Gool. Acquisition of detailed models for virtual reality. In J. Barcelo, M. Forte, and D. Sanders, editors, *Virtual Reality in Archaeology*, pages 71–77. ArcheoPress, 2000.

[Pollefeys *et al.*, 2000c] Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, Bert Deknuydt, and Luc Van Gool. Three-dimensional scene reconstruction from images. In *Proceedings of the SPIE Electronic Imaging, Three-Dimensional Image Capture and Applications III*, pages 215–226, 2000.

[Pollefeys *et al.*, 2000d] Marc Pollefeys, Maarten Vergauwen, and Luc Van Gool. Automatic 3d modeling from image sequences. In *International Archive of Photogrammetry and Remote Sensing*, volume XXXIII, pages 619–626, 2000.

[Pollefeys *et al.*, 2002a] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Kurt Cornelis, Frank Verbiest, and Jan Tops. Video-to-3d. In *Proceedings of Photogrammetric Computer Vision*, 2002.

[Pollefeys *et al.*, 2002b] Marc Pollefeys, Frank Verbiest, and Luc Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *Proceedings of the European Conference on Computer Vision*, pages 837–851, 2002.

[Pollefeys *et al.*, 2003] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Kurt Cornelis, Frank Verbiest, and Jan Tops. 3d capture of archaeology and architecture with a hand-held camera. In *Proceedings of the ISPRS Workshop on Vision Techniques for Digital Architectural and Archaeological Archives*, pages 262–267, 2003.

[Pollefeys *et al.*, 2004] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, and Jan Tops. Visual modeling with a hand-help camera. *International Journal of Computer Vision*, 59:207–232, 2004.

[Pritchett and Zisserman, 1998a] Philip Pritchett and Andrew Zisserman. Matching and reconstruction from widely separated views. In *SMILE*, pages 78–92, 1998.

[Pritchett and Zisserman, 1998b] Philip Pritchett and Andrew Zisserman. Wide baseline stereo matching. In *Proceedings of the International Conference on Computer Vision*, pages 754–760, 1998.

[Proietti and Faloutsos, 1998] Guido Proietti and Christos Faloutsos. Selectivity estimation of window queries for line segment datasets. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, pages 340–347, New York, NY, USA, 1998. ACM Press.

[Repko and Pollefeys, 2005] Jason Repko and Marc Pollefeys. 3d models from extended uncalibrated video sequences: Addressing key-frame selection and projective drift. In *Proc. of 3DIM*, 2005.

[Ripley, 1996] B. D. Ripley. *Pattern Recognition and Neural Networks.* Cambride University Press, 1996.

[Rissanen, 1985] Jorma Rissanen. Minimum-Description-Length Principle. In S.Kotz and N.L.Johnson, editors, *Encyclopedia of Statistical Science*, volume 5, pages 523–527. John Wiley & Sons, 1985.

[Rother and Carlsson, 2002] Carsten Rother and Stefan Carlsson. Linear multi view reconstruction and camera recovery using a reference plane. *International Journal of Computer Vision*, 49(2-3):117–141, 2002.

[Rother, 2003] Carsten Rother. Linear multi-view reconstruction of points, lines, planes and cameras using a reference plane. In *Proceedings of the International Conference on Computer Vision*, pages 1210–1217, 2003.

[Schaffalitzky and Zisserman, 1998] Frederik Schaffalitzky and Andrew Zisserman. Geometric grouping of repeated elements within images. In *Proceedings of the British Machine Vision Conference*, 1998.

[Schaffalitzky and Zisserman, 2000] Frederik Schaffalitzky and Andrew Zisserman. Planar grouping for automatic detection of vanishing lines and points. *Image Vision Comput.*, 18(9):647–658, 2000.

[Schaffalitzky and Zisserman, 2001] Frederik Schaffalitzky and Andrew Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proceedings of the International Conference on Computer Vision*, pages 636–643, 2001.

[Schaffalitzky and Zisserman, 2002] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Proceedings of European Conference on Computer Vision*, pages 414–431, 2002.

[Scharstein and Szeliski, 2002] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal on Computer Vision*, 47(1-3):7–42, 2002.

[Schmid and Mohr, 1997] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[Schmid and Zisserman, 1997] Cordelia Schmid and Andrew Zisserman. Automatic line matching across views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 666–671, 1997.

[Schmid and Zisserman, 2000] Cordelia Schmid and Andrew Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 40(3):199–233, 2000.

[Schmid *et al.*, 2000] Cordelia Schmid, Roger Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[Schuster, 2004] H.-F. Schuster. Segmentation of lidar data using the tensor voting framework. In *Proceedings of the 20th ISPRS Congress, Istanbul, Turkey*, 2004.

[Se *et al.*, 2005] S. Se, D. G. Lowe, and J. J. Little. Vision-Based Global Localization and Mapping for Mobile Robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

[Sester and Förstner, 1989] M. Sester and W. Förstner. Object Location Based on Uncertain Models. In *Proceedings of the DAGM Symposium, Hamburg*, 1989.

[Shafique and Shah, 2005] Khurram Shafique and Mubarak Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):51–65, 2005.

[Shannon, 1948] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[Shi and Liu, 2000] Wenzhong Shi and Wenbao Liu. A stochastic process-based model for the positional error of line segments in gis. *International Journal of Geographical Information Science*, 14(1):51–66, 2000.

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Shi and Tomasi, 1994] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 593 – 600, 1994.

[Shi, 1998] Wenzhong Shi. A generic statistical approach for modelling error of geometric features in gis. *International Journal of Geographical Information Science*, 12(2):131–143, 1998.

[Sim *et al.*, 2005] R. Sim, P. Elinas, M. Griffin, and J. J. Little. Vision-based SLAM using the Rao-Blackwellised particle filter. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR)*, pages 9–16, Edinburgh, Scotland, 2005.

[Skrypnyk and Lowe, 2004] Iryna Skrypnyk and David G. Lowe. Scene modelling, recognition and tracking with invariant image features. In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 110–119. IEEE Computer Society, 2004.

[Smith and Brady, 1997] S. M. Smith and J. M. Brady. SUSAN – A new approach to low level image processing. *Int. Journal of Computer Vision*, 23:45–78, 1997.

[Spreeuwers *et al.*, 1997] L.J. Spreeuwers, K. Schutte, and Z. Houkes. A model driven approach to extract buildings from multi-view aerial imagery. In A. Gruen, E.P. Baltsavias, and O. Henricsson, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, pages 109–118. Birkhäuser, Basel, 1997.

[Steffen, 2006] Richard Steffen. Zur Bestimmung der Trajektorie von Fremdfahrzeugen aus monokularen Bildfolgen. Technical report, Institute für Photogrammetrie, Universität Bonn, 2006.

[Stewenius *et al.*, 2005a] Henrik Stewenius, David Nister, Fredrik Kahl, and Frederik Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 789–794, San Diego, USA, 2005.

[Stewénius *et al.*, 2005b] Henrik Stewénius, Frederik Schaffalitzky, and David Nistér. How hard is 3-view triangulation really? In *Proceedings of the International Conference on Computer Vision*, pages 686–693, 2005.

[Stewenius *et al.*, 2006] Henrik Stewenius, Christopher Engels, and David Nister. Recent developments on direct relative orientation. *ISPRS Journal*, 2006. to appear.

[Stolfi, 1991] J. Stolfi. *Oriented Projective Geometry.* Academic Press, 1991.

[Tang and Medioni, 2002] C.K. Tang and G. Medioni. Curvature-augmented tensor voting for shape inference from noisy 3d data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):858–864, June 2002.

[Taylor and Kriegman, 1995] Camillo J. Taylor and David J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(11):1021–1032, 1995.

[Taylor *et al.*, 1996] Camillo J. Taylor, Paul E. Debevec, and Jitendra Malik. Reconstructing polyhedral models of architectural scenes from photographs. In *Proceedings of European Conference on Computer Vision*, pages 659–668, 1996.

[Tell and Carlsson, 2000] Dennis Tell and Stefan Carlsson. Wide baseline point matching using affine invariants computed from intensity profiles. In *Proceedings of European Conference on Computer Vision*, pages 814–828, London, UK, 2000. Springer-Verlag.

[Tell and Carlsson, 2002] Dennis Tell and Stefan Carlsson. Combining appearance and topology for wide baseline matching. In *Proceedings of European Conference on Computer Vision*, pages 68–81, 2002.

[Thormählen *et al.*, 2004] Thorsten Thormählen, Hellward Broszio, and Axel Weissenfeld. Keyframe selection for camera motion and structure estimation from multiple views. In *Proceedings of European Conference on Computer Vision*, pages 523–535, 2004.

[Torr and Zisserman, 1999] Philip H. S. Torr and Andrew Zisserman. Feature based methods for structure and motion estimation. In *Workshop on Vision Algorithms*, pages 278–294, 1999.

[Torr and Zisserman, 2000] P. H. S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *LNCS*, pages 278–294. Springer, 2000.

[Torr, 1997] P. H. S. Torr. An assessment of information criteria for motion model selection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, page 47, Washington, DC, USA, 1997. IEEE Computer Society.

[Torr, 2002] Philip H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002.

[Triggs *et al.*, 2000] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag.

[Triggs, 1995a] B. Triggs. Matching constraints and the joint image. In *Proceedings of the International Conference on Computer Vision*, page 338, Washington, DC, USA, 1995. IEEE Computer Society.

[Triggs, 1995b] Bill Triggs. The geometry of projective reconstruction i: Matching constraints and the joint image. Circulated in 1995. Accepted subject to revision to IJCV in 1995, but never completed, 1995.

[Triggs, 2004] Bill Triggs. Detecting keypoints with stable position, orientation, and scale under illumination changes. In *Proceedings of European Conference on Computer Vision*, pages 100–113, 2004.

[Tuytelaars and Gool, 2004] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal on Computer Vision*, 59(1):61–85, 2004.

[Utcke, 1998] Sven Utcke. Grouping based on projective geometry constraints and uncertainty. In *Proceedings of the International Conference on Computer Vision*, page 739, Washington, DC, USA, 1998. IEEE Computer Society.

[Van Gool *et al.*, 1998] L. Van Gool, M. Proesmans, and A. Zisserman. Planar homologies as a basis for grouping and recognition. *Image and Vision Computing*, 16:21–26, January 1998.

[Vedaldi and Soatto, 2005] Andrea Vedaldi and Stefano Soatto. Features for recognition: Viewpoint invariance for non-planar scenes. In *Proceedings of the International Conference on Computer Vision*, pages 1474–1481, 2005.

[Vedaldi *et al.*, 2005] Andrea Vedaldi, Hailin Jin, Paolo Favaro, and Stefano Soatto. Kalmansac: Robust filtering by consensus. In *Proceedings of the International Conference on Computer Vision*, pages 633–640, 2005.

[Venkateswar and Chellappa, 1995] V. Venkateswar and R. Chellappa. Hierarchical stereo and motion correspondence using feature groupings. *IJCV*, 15(3):245–269, July 1995.

[Vitanyi, 2004] Paul M.B. Vitanyi. Algorithmic statistics and kolmogorov's structure functions. In P.M.B. Vitanyi, editor, *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2004.

[Vosselman *et al.*, 2004] George Vosselman, Monika Sester, and Helmut Meyer. Basic computer vision techniques. In J.C.McGlone, E.M.Mikhail, and J.Bethel, editors, *Manual of Photogrammetry*, pages 455–504. ASPRS, 2004.

[Vosselman, 1992] G. Vosselman. *Relational Matching*, volume 628 of *Lecture Notes in Computer Science*. Springer, 1992.

[Welch and Bishop, 1995] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[Werner and Pajdla, 2001a] Tomás Werner and Tomás Pajdla. Cheirality in epipolar geometry. In *Proceedings of the International Conference on Computer Vision*, pages 548–553, 2001.

[Werner and Pajdla, 2001b] Tomáš Werner and Tomáš Pajdla. Oriented matching constraints. In T Cootes and C Taylor, editors, *British Machine Vision Conference 2001*, pages 441–450, London, UK, September 2001. British Machine Vision Association.

[Werner and Zisserman, 2002a] T. Werner and A. Zisserman. Model selection for automated architectural reconstruction from multiple views. In *Proceedings of the British Machine Vision Conference*, pages 53–62, 2002.

[Werner and Zisserman, 2002b] Tomás Werner and Andrew Zisserman. New techniques for automated architectural reconstruction from photographs. In *Proceedings of European Conference on Computer Vision*, pages 541–555, 2002.

[Werner *et al.*, 2001] T. Werner, F. Schaffalitzky, and A. Zisserman. Automated architecture reconstruction from close-range photogrammetry. In *Proceedings on CIPA 2001 International Symposium: Surveying and Documentation of Historic Buildings – Monuments – Sites, Traditional and Modern Methods*, September 2001.

[Werner, 2003] Tomás Werner. Constraint on five points in two images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 203–208, 2003.

[Xiao and Shah, 2003] Jiangjian Xiao and Mubarak Shah. Two-frame wide baseline matching. In *Proceedings of the International Conference on Computer Vision*, pages 603–609, 2003.

[Zhang and Faugeras, 1994] Zhengyou Zhang and Olivier Faugeras. Finding planes and clusters of objects from 3d line segments with application to 3d motion estimation. *CVGIP: Image Understanding*, 60(3):267–284, 1994.

[Zhang, 1994] Zhengyou Zhang. Token tracking in a cluttered scene. *International Journal of Image and Vision Computing*, 12(2):110–120, 1994.

[Zhang, 1995] Zhengyou Zhang. Estimating motion and structure from correspondences of line segments between two perspective images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1129–1139, 1995.

[Zhang, 1998] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. *Int. J. Comput. Vision*, 27(2):161–195, 1998.

[Zhang, 1999] Zhengyou Zhang. Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. In *Proceedings of the International Conference on Computer Vision*, pages 666–673, Corfu, Greece, 1999.

[Zisserman *et al.*, 1995] A. Zisserman, J. Mundy, D. Forsyth, J. Liu, N. Pillow, C. Rothwell, and S. Utcke. Class-based grouping in perspective images. In *Proceedings of the International Conference on Computer Vision*, 1995.